Batch NFS

D. J. Bernstein
University of Illinois at Chicago &
Technische Universiteit Eindhoven

Tanja Lange Technische Universiteit Eindhoven

In this talk log *L* means $(1 + o(1))(\log N)^{1/3}(\log \log N)^{2/3}$. *L* is often written " $L_N(1/3)$ " or " $L_N(1/3)^{1+o(1)}$ ".

Exponents of L in this talk are limited to 10^{-6} **Z**.

Rigorously proven? Ha ha ha.

2003 Shamir–Tromer, 2003 Lenstra-Tromer-Shamir-Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann-Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing $<10^9$ dollars.

Batch NFS

D. J. BernsteinUniversity of Illinois at Chicago &Technische Universiteit Eindhoven

Tanja Lange Technische Universiteit Eindhoven

In this talk log *L* means $(1 + o(1))(\log N)^{1/3}(\log \log N)^{2/3}$. *L* is often written "*L*_N(1/3)" or "*L*_N(1/3)^{1+o(1)"}.

Exponents of L in this talk are limited to 10^{-6} **Z**.

Rigorously proven? Ha ha ha.

2003 Shamir–Tromer, 2003 Lenstra-Tromer-Shamir-Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann-Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing $<10^9$ dollars. So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Batch NFS

D. J. BernsteinUniversity of Illinois at Chicago &Technische Universiteit Eindhoven

Tanja Lange Technische Universiteit Eindhoven

In this talk log *L* means $(1 + o(1))(\log N)^{1/3}(\log \log N)^{2/3}$. *L* is often written "*L*_N(1/3)" or "*L*_N(1/3)^{1+o(1)"}.

Exponents of *L* in this talk are limited to 10^{-6} **Z**.

Rigorously proven? Ha ha ha.

2003 Shamir–Tromer, 2003 Lenstra-Tromer-Shamir-Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann-Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing $<10^9$ dollars. So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right? Wrong!

FS

rnstein

- ty of Illinois at Chicago & che Universiteit Eindhoven
- ange
- che Universiteit Eindhoven
- alk log L means)) $(\log N)^{1/3}(\log \log N)^{2/3}$.
- n written
- 3)" or " $L_N(1/3)^{1+o(1)}$ ".
- its of L in this talk ed to 10^{-6} **Z**.
- sly proven? Ha ha ha.

2003 Shamir–Tromer, 2003 Lenstra-Tromer-Shamir-Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann-Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing $<10^9$ dollars.

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example dnssecis signed is at Chicago & siteit Eindhoven

siteit Eindhoven

neans ^{/3}(log log N)^{2/3}.

 $(1/3)^{1+o(1)}$ ".

this talk ⁶**Z**.

? Ha ha ha.

2003 Shamir–Tromer, 2003 Lenstra–Tromer–Shamir– Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann– Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing <10⁹ dollars.

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP dnssec-deployme is signed by an RS

ago & hoven

hoven

 $N)^{2/3}$.

(1)"

а.

2003 Shamir–Tromer, 2003 Lenstra–Tromer–Shamir– Kortsmit–Dodson–Hughes– Leyland, 2005 Geiselmann– Shamir–Steinwandt–Tromer, 2005 Franke–Kleinjung–Paar–Pelzl– Priplata–Stahlke, etc.: RSA-1024 is breakable in a year by an attack machine costing <10⁹ dollars.

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 ke

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

Most "DNSSEC" signatures follow a similar pattern.

So the Internet switched to RSA-2048, and we no longer care about RSA-1024 security, right?

Wrong!

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.

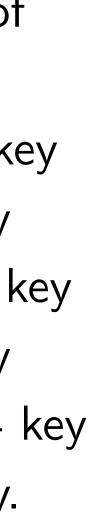
amir-Tromer, 2003 -Tromer–Shamir– t–Dodson–Hughes– 2005 Geiselmann-Steinwandt–Tromer, 2005 Kleinjung–Paar–Pelzl– -Stahlke, etc.: RSA-1024 able in a year by an attack costing $< 10^9$ dollars.

nternet switched to 18, and we no longer care SA-1024 security, right?

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.



Users se 1. "The more that 2. "The off-the-s attacker 3. For s switch k the atta

ner, 2003

- hamir–
- -Hughes-
- selmann–
- lt–Tromer, 2005
- -Paar-Pelzl-
- etc.: RSA-1024
- ear by an attack (10⁹ dollars.
- vitched to e no longer care security, right?

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.

Users seem uncon 1. "The attack m more than this RS 2. "The attack m off-the-shelf; it's c attackers building 3. For signatures: switch keys every the attack machin

, 2005 :I– -1024 attack

ſS.

r care ght? Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.

Users seem unconcerned:

- 1. "The attack machine cos
- more than this RSA key is w
- 2. "The attack machine isn" off-the-shelf; it's only for
- attackers building ASICs."
- 3. For signatures: "We
- switch keys every month, an
- the attack machine takes a

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.

Users seem unconcerned: 1. "The attack machine costs more than this RSA key is worth." 2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs." 3. For signatures: "We switch keys every month, and the attack machine takes a year."

Example: The IP address of dnssec-deployment.org is signed by an RSA-1024 key signed by an RSA-2048 key signed by org's RSA-1024 key signed by an RSA-2048 key signed by a root RSA-1024 key signed by an RSA-2048 key.

Most "DNSSEC" signatures follow a similar pattern.

Another example: SSL has used many millions of RSA-1024 keys. Imagine that an attacker has recorded tons of SSL traffic.

Users seem unconcerned: 1. "The attack machine costs 2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs." 3. For signatures: "We switch keys every month, and the attack machine takes a year." Real quote: "DNSSEC signing keys should be large enough to avoid all known cryptographic attacks during the effectivity

period of the key."

- more than this RSA key is worth."

e: The IP address of -deployment.org I by an RSA-1024 key y an RSA-2048 key y org's RSA-1024 key y an RSA-2048 key y a root RSA-1024 key y an RSA-2048 key.

NSSEC" signatures similar pattern.

example: SSL has used illions of RSA-1024 keys. that an attacker has tons of SSL traffic. Users seem unconcerned:

"The attack machine costs more than this RSA key is worth."
 "The attack machine ice".

2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs."

3. For signatures: "We switch keys every month, and the attack machine takes a year."

Real quote: "DNSSEC signing keys should be large enough to avoid all known cryptographic attacks during the effectivity period of the key."

Continua despite l broken a fact, the estimate of a 700 breaking would no amounts power in be detec single ke estimate safely us least the

address of ent.org 5A-1024 key -2048 key 5A-1024 key -2048 key SA-1024 key -2048 key.

signatures ttern.

SSL has used RSA-1024 keys. ttacker has SL traffic. Users seem unconcerned:

1. "The attack machine costs more than this RSA key is worth." 2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs." 3. For signatures: "We switch keys every month, and the attack machine takes a year." Real quote: "DNSSEC signing

Real quote: "DNSSEC signing keys should be large enough to avoid all known cryptographic attacks during the effectivity period of the key."

Continuation of qu despite huge effort broken a regular 1 fact, the best com estimated to be th of a 700-bit key. A breaking a 1024-b would need to exp amounts of netwo power in a way the be detected in ord single key. Becaus estimated that mo safely use 1024-bit least the next ten

ЭУ

ey

key

used keys.

S

Users seem unconcerned:

1. "The attack machine costs more than this RSA key is worth."

2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs."

3. For signatures: "We switch keys every month, and the attack machine takes a year."

Real quote: "DNSSEC signing keys should be large enough to avoid all known cryptographic attacks during the effectivity period of the key."

Continuation of quote: "To despite huge efforts, no one broken a regular 1024-bit ke fact, the best completed att estimated to be the equivale of a 700-bit key. An attacke breaking a 1024-bit signing would need to expend pheno amounts of networked comp power in a way that would r be detected in order to brea single key. Because of this, estimated that most zones of safely use 1024-bit keys for least the next ten years."

Users seem unconcerned:

1. "The attack machine costs more than this RSA key is worth."

2. "The attack machine isn't off-the-shelf; it's only for attackers building ASICs."

3. For signatures: "We switch keys every month, and the attack machine takes a year."

Real quote: "DNSSEC signing keys should be large enough to avoid all known cryptographic attacks during the effectivity period of the key."

Continuation of quote: "To date, despite huge efforts, no one has broken a regular 1024-bit key; in fact, the best completed attack is estimated to be the equivalent of a 700-bit key. An attacker breaking a 1024-bit signing key would need to expend phenomenal amounts of networked computing power in a way that would not be detected in order to break a single key. Because of this, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years."

em unconcerned:

attack machine costs an this RSA key is worth."

attack machine isn't

helf; it's only for

s building ASICs."

ignatures: "We

eys every month, and

ck machine takes a year."

ote: "DNSSEC signing ould be large enough to known cryptographic during the effectivity f the key." Continuation of quote: "To date, despite huge efforts, no one has broken a regular 1024-bit key; in fact, the best completed attack is estimated to be the equivalent of a 700-bit key. An attacker breaking a 1024-bit signing key would need to expend phenomenal amounts of networked computing power in a way that would not be detected in order to break a single key. Because of this, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years."

Goal of analyze specifica *ratio*, of "Many" "Price-p area-tin "RAM" bit integ accessin realistic; "Asymp suppress speedup

cerned:

achine costs

A key is worth."

achine isn't

only for

ASICs."

"We

month, and

e takes a year."

SEC signing ge enough to yptographic

effectivity

Continuation of quote: "To date, despite huge efforts, no one has broken a regular 1024-bit key; in fact, the best completed attack is estimated to be the equivalent of a 700-bit key. An attacker breaking a 1024-bit signing key would need to expend phenomenal amounts of networked computing power in a way that would not be detected in order to break a single key. Because of this, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years."

Goal of our "Batc analyze the asymp specifically price-p *ratio*, of breaking "Many": e.g. mill "Price-performanc area-time produc "RAM" metric (ad bit integers has sa accessing array of realistic; "AT" me "Asymptotic": We suppress polynomi speedups are supe

ts vorth." 't

d year."

ng to

ic

/

Continuation of quote: "To date, despite huge efforts, no one has broken a regular 1024-bit key; in fact, the best completed attack is estimated to be the equivalent of a 700-bit key. An attacker breaking a 1024-bit signing key would need to expend phenomenal amounts of networked computing power in a way that would not be detected in order to break a single key. Because of this, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years."

Goal of our "Batch NFS" pa analyze the asymptotic cost specifically price-performance ratio, of breaking many RSA "Many": e.g. millions. "Price-performance ratio": area-time product for chips "RAM" metric (adding two bit integers has same cost a accessing array of size 2^{64}) realistic; "AT" metric is rea "Asymptotic": We systemat suppress polynomial factors. speedups are superpolynomi

Continuation of quote: "To date, despite huge efforts, no one has broken a regular 1024-bit key; in fact, the best completed attack is estimated to be the equivalent of a 700-bit key. An attacker breaking a 1024-bit signing key would need to expend phenomenal amounts of networked computing power in a way that would not be detected in order to break a single key. Because of this, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years."

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys. "Many": e.g. millions. "Price-performance ratio": area-time product for chips. "RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic. "Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

ation of quote: "To date, nuge efforts, no one has regular 1024-bit key; in e best completed attack is d to be the equivalent -bit key. An attacker a 1024-bit signing key eed to expend phenomenal s of networked computing a way that would not ted in order to break a ey. Because of this, it is d that most zones can se 1024-bit keys for at e next ten years."

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically price-performance ratio, of breaking many RSA keys. "Many": e.g. millions.

"Price-performance ratio": area-time product for chips. "RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic.

"Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

Best res time L^{1} . using ch AT is L^{-}

lote: "To date, s, no one has 024-bit key; in pleted attack is ne equivalent An attacker it signing key end phenomenal rked computing at would not er to break a e of this, it is st zones can t keys for at years."

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys. "Many": e.g. millions. "Price-performance ratio": area-time product for chips. "RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic. "Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

Best result known time $L^{1.185632}$ using chip area L^0 AT is $L^{1.976052}$.

date, has y; in ack is ent er key omenal uting not k a it is an at

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys. "Many": e.g. millions. "Price-performance ratio": area-time product for chips. "RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic.

"Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

time $L^{1.185632}$ AT is $L^{1.976052}$.

Best result known for one ke

using chip area $L^{0.790420}$:

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys.

"Many": e.g. millions.

"Price-performance ratio": area-time product for chips.

"RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic.

"Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$;

AT is $L^{1.976052}$

Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys.

"Many": e.g. millions.

"Price-performance ratio": area-time product for chips.

"RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic.

"Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

Best result known for *one* key: time $L^{1.185632}$

using chip area $L^{0.790420}$; AT is $L^{1.976052}$

Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$

using chip area $L^{1.181600}$; AT per key is $L^{1.704000}$.



Goal of our "Batch NFS" paper: analyze the *asymptotic* cost, specifically *price-performance* ratio, of breaking many RSA keys.

"Many": e.g. millions.

"Price-performance ratio": area-time product for chips.

"RAM" metric (adding two 64bit integers has same cost as accessing array of size 2^{64}) is not realistic; "AT" metric is realistic.

"Asymptotic": We systematically suppress polynomial factors. Our speedups are superpolynomial.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$; AT is $L^{1.976052}$ Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$ using chip area $L^{1.181600}$; AT per key is $L^{1.704000}$. This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would

guess from 1982 Pomerance.

our "Batch NFS" paper: the *asymptotic* cost, lly price-performance breaking many RSA keys.

: e.g. millions.

erformance ratio": ne product for chips.

metric (adding two 64ers has same cost as g array of size 2⁶⁴) is not "AT" metric is realistic.

totic": We systematically polynomial factors. Our s are superpolynomial.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$: AT is $L^{1.976052}$.

Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$ using chip area $L^{1.181600}$: *AT* per key is $L^{1.704000}$.

This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would guess from 1982 Pomerance.



Asympto 1. Attac is reduce can targ 2. Prima memory for off-tl 3. Attac (and car breaking

h NFS" paper: *ototic* cost, *erformance many* RSA keys.

e ratio":

t for chips.

dding two 64-

me cost as size 2⁶⁴) is not etric is realistic.

e systematically al factors. Our rpolynomial. Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$; AT is $L^{1.976052}$.

Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$ using chip area $L^{1.181600}$; AT per key is $L^{1.704000}$.

This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would guess from 1982 Pomerance.

Asymptotic consec

- 1. Attack cost per
- is reduced, so atta
- can target lower-v

2. Primary bottler memory factorizat for off-the-shelf gr

Attack time is
 (and can be reduc
 breaking key rotat

```
aper:
e
A keys.
S.
64-
S
is not
listic.
cically
 Our
```

al.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$; AT is $L^{1.976052}$. Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$

using chip area $L^{1.181600}$: AT per key is $L^{1.704000}$.

This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would guess from 1982 Pomerance.

Asymptotic consequences:

- 1. Attack cost per key
- is reduced, so attacker
- can target lower-value keys.
- 2. Primary bottleneck is low
- memory factorization—well
- for off-the-shelf graphics car
- 3. Attack time is reduced
- (and can be reduced more), breaking key rotation.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$; AT is $L^{1.976052}$

Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$ using chip area $L^{1.181600}$; AT per key is $L^{1.704000}$.

This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would guess from 1982 Pomerance.

Asymptotic consequences:

1. Attack cost per key is reduced, so attacker can target lower-value keys.

2. Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

3. Attack time is reduced (and can be reduced more), breaking key rotation.

Best result known for *one* key: time $L^{1.185632}$ using chip area $L^{0.790420}$; AT is $L^{1.976052}$

Our main result for a batch of $L^{0.5}$ keys: time $L^{1.022400}$ using chip area $L^{1.181600}$; AT per key is $L^{1.704000}$.

This paper also looks more closely at $L^{o(1)}$, analyzing asymptotic speedup from early-abort ECM. Results are not what one would guess from 1982 Pomerance.

Asymptotic consequences:

1. Attack cost per key is reduced, so attacker can target lower-value keys.

2. Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

3. Attack time is reduced (and can be reduced more), breaking key rotation.

but no basis for confidence.

- "Do the asymptotics really kick in before 1024 bits?" — Maybe not,

ult known for *one* key: 185632

ip area $L^{0.790420}$: L.976052

n result for of $L^{0.5}$ keys: 022400

ip area $L^{1.181600}$; key is $L^{1.704000}$.

per also looks more closely , analyzing asymptotic from early-abort ECM. are not what one would om 1982 Pomerance.

Asymptotic consequences:

1. Attack cost per key is reduced, so attacker can target lower-value keys.

2. Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

3. Attack time is reduced (and can be reduced more), breaking key rotation.

"Do the asymptotics really kick in before 1024 bits?" — Maybe not, but no basis for confidence.

Eratosth

Sieving s using pr

	L 1] 1 1]] 1] 1 1	1				
y	20	4 5 6 7	L3 [4	1 1 2 3		8 Q	5 6 7	5 4 5	1 2 3 4 5 6 7 8 9 0
	2	2	2	2	2	2	2	2	2
		2		2		2		2	
		2				2			
		2							
	3	3	2	3	0	3	3	J	2

for *one* key: .790420. r ys: .181600. 04000

oks more closely g asymptotic y-abort ECM. hat one would Pomerance. Asymptotic consequences:

Attack cost per key
 is reduced, so attacker
 can target lower-value keys.

 Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

 Attack time is reduced (and can be reduced more), breaking key rotation.

"Do the asymptotics really kick in before 1024 bits?" — Maybe not, but no basis for confidence.

Eratosthenes for s

Sieving small integrusing primes 2, 3, 5

1 2 3 4 5 6 7 8 9	2	2	
5 4 5	22	3	-
5 6	2	3	5
/ 8	222		(
9 10	2	33	5
10 11 12 13 14 15 16	22	3	
13 14	2		7
15 16	222	3	5
17			
18 19 20	2	33	
20	22		5

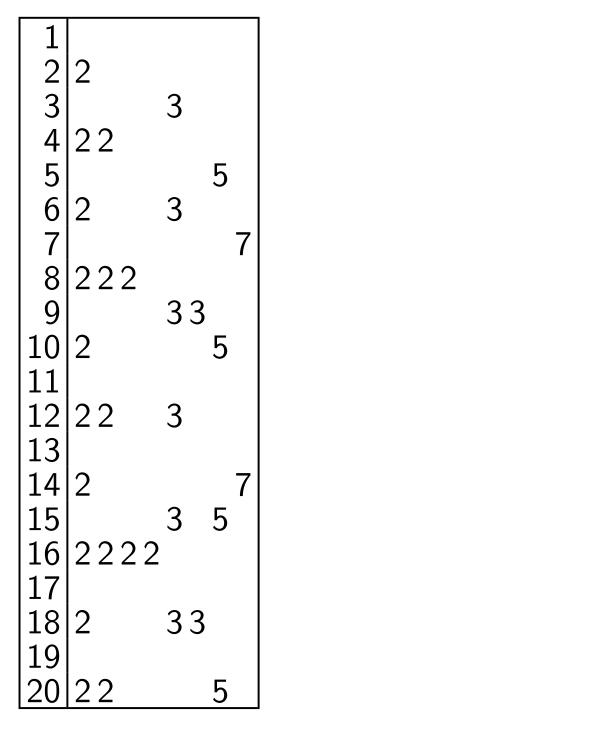
Asymptotic consequences:

1. Attack cost per key is reduced, so attacker can target lower-value keys.

2. Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

3. Attack time is reduced (and can be reduced more), breaking key rotation.

"Do the asymptotics really kick in before 1024 bits?" — Maybe not, but no basis for confidence.



etc.

closely tic CM. bluc

Eratosthenes for smoothness

Sieving small integers i > 0using primes 2, 3, 5, 7:

Asymptotic consequences:

1. Attack cost per key is reduced, so attacker can target lower-value keys.

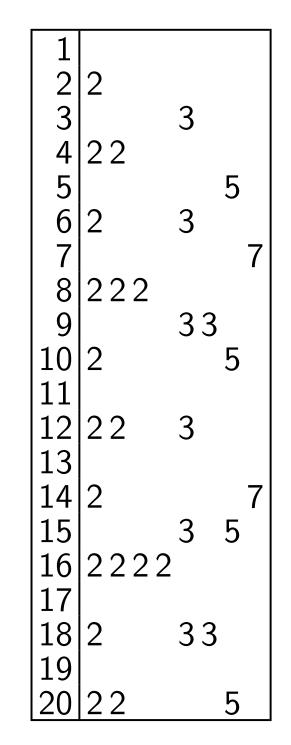
2. Primary bottleneck is lowmemory factorization—well suited for off-the-shelf graphics cards.

3. Attack time is reduced (and can be reduced more), breaking key rotation.

"Do the asymptotics really kick in before 1024 bits?" — Maybe not, but no basis for confidence.

Eratosthenes for smoothness

Sieving small integers i > 0using primes 2, 3, 5, 7:



otic consequences:

- k cost per key ed, so attacker
- et lower-value keys.
- ary bottleneck is lowfactorization—well suited ne-shelf graphics cards.
- k time is reduced
- n be reduced more),
- key rotation.
- asymptotics really kick in 024 bits?" — Maybe not, basis for confidence.

Eratosthenes for smoothness

Sieving small integers i > 0using primes 2, 3, 5, 7:

	-		
$1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16$	2	2	
3 4	22	3	_
5 6	2	3	5
7 8	222)	7
9 10	2	33	5
11 12	22	3	
13 14	2	0	7
14	2	3	5
17			
18 19	2 22	33	
20	22		5

etc.

The **Q** s

Sieving using pr

20	18 19 20	15 16 17	13 14 15	11 12 13	10	8	6 7	1 2 3 4 5 6 7 8 9	1 2
2	2	2	2	2	2	2	2	2	2
2		2		2		2		2	
		2				2			
		2							
	3	J	3	3	5	2	3	3	

quences:

- r key
- cker
- alue keys.
- neck is low-
- ion—well suited
- aphics cards.
- reduced
- ed more),
- ion.
- ics really kick in — Maybe not,
- onfidence.

Eratosthenes for smoothness

Sieving small integers i > 0using primes 2, 3, 5, 7:

	1		
$ \begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16\end{array} $	2	2	
3 4 5	22	3	
5 6 7	2	5 3	7
8	222	33	
9 10 11	2	55	;
11 12	22	3	
13 14 15	2	3 5	7
16 17	2222		,
18 10	2 22	33	
20	22	5	



The **Q** sieve

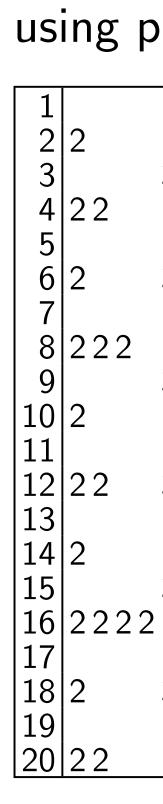
Sieving *i* and 611 using primes 2, 3, 5

1				612	2
1 2 3 4 5 6 7 8 9	2			613	
3		3		614	2
4	22			615	
5		_	5	616	2
6	2	3		617	
7			7	618	2
8	222			619	
9	_	33		620	2
10	2		5	621	
11		-		622	2
12	22	3		623	
13			_	624	2
14	2	-	_ 7	625	
15		3	5	626	2
16	222	2		627	
17				628	2
18	2	33	3	629	
18 19			_	629 630 631	2
20	22		5	631	

Eratosthenes for smoothness

Sieving small integers i > 0using primes 2, 3, 5, 7:

123456789	0							
2	2				2			
3		~			3			
4	2	2					_	
5					~		5	
6	2				3			
7		_	_					7
8	2	2	2		_	_		
9					3	3		
10 11	2						5	
11								
12 13	2	2			3			
13								
14	2							7
15					3		5	
16	2	2	2	2				
17								
18	2				3	3		
19								
18 19 20	2	2					5	



etc.

/-

suited

kick in

e not,

ds.

etc.

The **Q** sieve

Sieving *i* and 611 + i for sm using primes 2, 3, 5, 7:

		_									
			612	2	2			3	3		
			613								
3			614	2							
			615					3			5
	5		616	2	2	2					
3			617								
	7		618	2				3			
			619								
33			620	2	2						5
	5		621					3	3	3	
			622	2							
3			623								
			624	2	2	2	2	3			
_	7		625								5
3	5		626	2				_			
			627					3			
			628		2						
33			629 630 631								
			630	2				3	3		5
	5		631								

Eratosthenes for smoothness

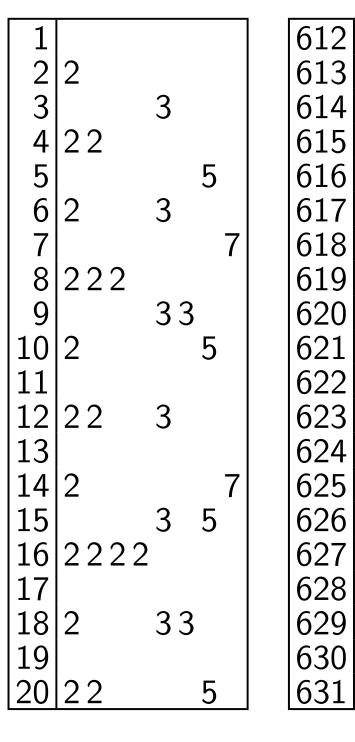
Sieving small integers i > 0using primes 2, 3, 5, 7:

1				
1 2 3 4 5 6 7 8 9	2			
3			3	
4	22			
5				5
6	2		3	
7				7
8	222	2	~ ~	
9	•		33	_
10	2			5
			2	
12	22		3	
13 1 4	0			_
14 15	2		2	7
$10\\11\\12\\13\\14\\15\\16$	222	า า	3	5
17		<u> </u>		
	2		33	
10 10	2		JJ	
20	2 22			5
20				5

etc.

The **Q** sieve

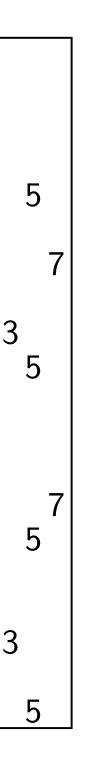
Sieving *i* and 611 + i for small *i* using primes 2, 3, 5, 7:



2	2			3	3						
2				3			5				
2	2	2		S			5				7
2				3							
2	2			2	о [,]		5				
2				2	3 3	S					7
2	2	2	2	3			F	F	5	F	7
2				3			5	5	5	5	
2	2			J							
2				3	3		5				7

enes for smoothness

small integers i > 0 imes 2, 3, 5, 7:



The **Q** sieve

Sieving *i* and 611 + *i* for small *i* using primes 2, 3, 5, 7:

_				_							
	123456789 10				612	2	2		3	3	
	2	2			613						
	3		3		614	2					
	4	22			615				3		
	5		5		616	2	2	2			
	6	2	3		617						
	7		7		618	2			3		
	8	222			619						
	9		33		620	2	2				
	10	2	5		621				3	3	3
	11				622	2					
	12	22	3		623						
	13					2	2	2 2	3		
	14	2	7		625						
	15		3 5		626	2					
	16	2222			627				3		
	17				628	2	2				
		2	33								
	18 19	_			629 630	2			3	3	
	20	22	5		631				-	-	

etc.



Have co the cong for some $14 \cdot 625$ 64 · 675 75 · 686 14 · 64 · $= 2^8 3^4 5$ gcd{611 = 47. 611 = 4

moothness

gers i > 05, 7:

The **Q** sieve

Sieving *i* and 611 + *i* for small *i* using primes 2, 3, 5, 7:

					<u> </u>								
1				612	2	2			3	3			
2	2			613									
3		3		614	2								
4	22			615					3		5		
5			5	616	2	2	2						7
6	2	3		617									
2 3 4 5 6 7		•	7	618	2				3				
8	222		-	619					•				
8 9		33		620	2	2					5		
10	2		5	621		_			3	33	0		
11		·	Ŭ	622	2				Ŭ				
12	22	3		623									7
13	~ ~	0		624	2	2	2	2	ર				•
14	2		7	625		2	2	2	0		55	55	
15	2	3	5	626	2						55	55	
$10 \\ 16$	2222		5	627					3				
17				628	2	\mathbf{c}			J				
	2	33				Ζ							
18 19	2	55		629	2				2	2	F		7
	2.2			630	2				3	S	5		1
20	22		5	631									

etc.

Have complete fac the congruences *i* for some *i*'s.

- $14 \cdot 625 = 2^1 3^0 5^4 7$
- $64 \cdot 675 = 2^6 3^3 5^2 7$ 75 \cdot 686 = 2^1 3^1 5^2 7
- $14 \cdot 64 \cdot 75 \cdot 625 \cdot 64 = 2^8 3^4 5^8 7^4 = (2^4)^{10}$
- $gcd\{611, 14 \cdot 64 \cdot 7 = 47.$
- $611 = 47 \cdot 13.$

The **Q** sieve

Sieving *i* and 611 + i for small *i* using primes 2, 3, 5, 7:

1				612	2	2			3	3			
2	2			613									
3		3		614	2								
4	22			615					3		5		
5			5	616	2	2	2						7
6	2	3		617									
1 2 3 4 5 6 7 8			7	618	2				3				
8	222		_	619					-				
9		33		620	2	2					5		
10	2	•••	5	621	_	_			3	33	Ū		
11	-		•	622	2				Ŭ	00			
12	22	3		623	_								7
13		0		624	2	2	2	2	ર				'
14	2		7	625	2	2	2	~	5		ቫ	55	5
15		3	5	626	2						5	55	5
16	2222		5	627	2				3				
17		•		628	2	\mathbf{O}			5				
	 	33				2							
	2	55		629 630					2	2	F		_
19			-	$\left \begin{array}{c} 030 \\ 031 \end{array} \right $	2				3	3	5		(
20	22		5	631									

etc.

- for some *i*'s.
- $14 \cdot 625 = 2^1 3^0 5^4 7^1.$
- $64 \cdot 675 = 2^6 3^3 5^2 7^0$
- $75 \cdot 686 = 2^1 3^1 5^2 7^3$.
- $14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$ $= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2.$
- $gcd{611, 14 \cdot 64 \cdot 75 2^43^2}$ = 47.
- $611 = 47 \cdot 13$.

Have complete factorization the congruences $i \equiv 611 + i$

The **Q** sieve

Sieving *i* and 611 + i for small *i* using primes 2, 3, 5, 7:

	1														
1				612	2	2			3	3					
2	2			613											
3		3		614	2										
4	22			615					3		5				
5			5	616	2	2	2		•		•				7
1 2 3 4 5 6 7 8 9	2	3	0	617		_	-								•
		0	7	618	2				3						
	222		1	619	2				5						
		22	>	620	2	\mathbf{c}					5				
		33			Ζ	Ζ			2	~ ~	-				
10	2		5	621	0				3	33					
11		0		622	2										_
12	22	3		623	_	_	-	_	_						1
13				624	2	2	2	2	3						
14	2		7	625							5	5	5	5	
15		3	5	626	2										
16	2222)		627					3						
17				628	2	2									
18	2	33	3	629											
19			-	630	2				3	3	5				7
20	22		5	631	-					`	0				•

Have complete factorization of the congruences $i \equiv 611 + i$ for some *i*'s.

- $14 \cdot 625 = 2^1 3^0 5^4 7^1$. $64 \cdot 675 = 2^6 3^3 5^2 7^0$ $75 \cdot 686 = 2^1 3^1 5^2 7^3$
- $14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$ $= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2.$
- $gcd{611, 14 \cdot 64 \cdot 75 2^4 3^2 5^4 7^2}$ = 47.

 $611 = 47 \cdot 13$.

ieve

i and 611 + *i* for small *i* imes 2, 3, 5, 7:

		12	2	2			3	3						
		13												
	6	14	2											
	6	15					3			5				
5	6	16	2	2	2									7
	6	17												
7	6	18	2				3							
		19												
3		20	2	2						5				
5		21					3	3	3	Ŭ				
0		22	2				U	U						
		23	2											7
		24	2	2	2	2	2							1
7		25	2	2	2	2	5			۲	Б	5	ፍ	
ו ה			\mathbf{c}							5	5	5	5	
5			2				\mathbf{c}							
		27	0	~			3							
•		28	2	2										
3	62	29 30 31	-				-	-		_				
	6	30	2				3	3		5				7
5	6	31												

Have complete factorization of the congruences $i \equiv 611 + i$ for some *i*'s.

 $14 \cdot 625 = 2^{1}3^{0}5^{4}7^{1}.$ $64 \cdot 675 = 2^{6}3^{3}5^{2}7^{0}.$ $75 \cdot 686 = 2^{1}3^{1}5^{2}7^{3}.$

 $14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$ = $2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2$.

 $gcd \{ 611, 14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2 \} = 47.$

 $611 = 47 \cdot 13.$

Ϊ 2ς472ι

The nun Generali $\rightarrow a \equiv a$ $\rightarrow a - b$ for root of nonze For any so that t

produces

Optimal $(\mu + o(1$

_			- 0	r	S	m	а		i		
D , 2	7		3	3							
						_					
2	2		3			5				7	
			3								
2			3	3	3	5					
2	2	2	3			5	5	5	5	7	
2			3								
			3	3		5				7	

Have complete factorization of the congruences $i \equiv 611 + i$ for some *i*'s.

 $14 \cdot 625 = 2^{1}3^{0}5^{4}7^{1}.$ $64 \cdot 675 = 2^{6}3^{3}5^{2}7^{0}.$ $75 \cdot 686 = 2^{1}3^{1}5^{2}7^{3}.$

 $14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$ $= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2.$

 $gcd\{611, 14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2\} = 47.$

 $611 = 47 \cdot 13.$

The number-field

Generalize $i \equiv i +$ $\rightarrow a \equiv a + bN$ (

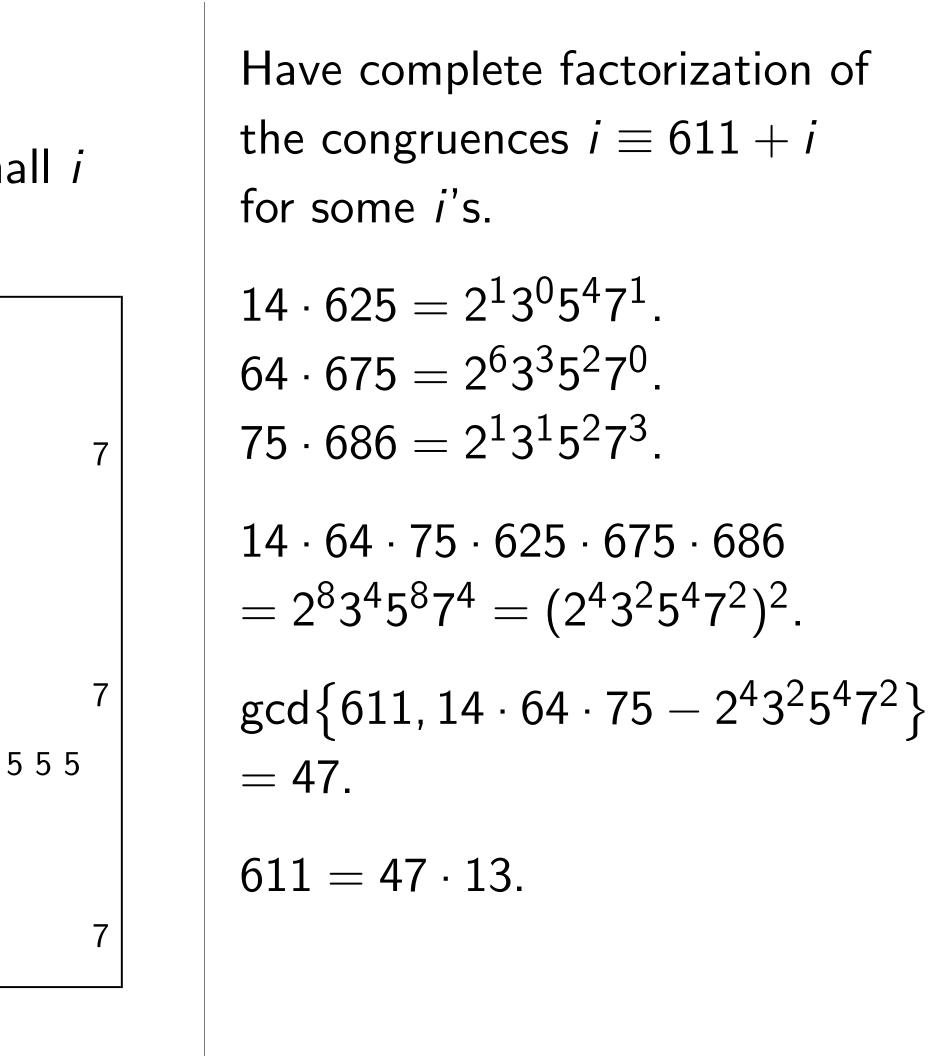
 $\rightarrow a - bm \equiv a - bc$

for root $\alpha \in \mathbf{C}$

of nonzero integer

For any *m* can find so that factoring *r* produces factoriza

Optimal choice of $(\mu + o(1))(\log N)^2$



The number-field sieve Generalize $i \equiv i + N$ (mod $\rightarrow a \equiv a + bN \pmod{N}$ $\rightarrow a - bm \equiv a - b\alpha \pmod{mod}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly. For any *m* can find α so that factoring m-lphaproduces factorization of N. Optimal choice of log *m* is

 $(\mu + o(1))(\log N)^{2/3}(\log \log N)$

Have complete factorization of the congruences $i \equiv 611 + i$ for some *i*'s.

 $14 \cdot 625 = 2^{1}3^{0}5^{4}7^{1}$ $64 \cdot 675 = 2^6 3^3 5^2 7^0$ $75 \cdot 686 = 2^1 3^1 5^2 7^3$

 $14 \cdot 64 \cdot 75 \cdot 625 \cdot 675 \cdot 686$ $= 2^8 3^4 5^8 7^4 = (2^4 3^2 5^4 7^2)^2.$

 $gcd\{611, 14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2\}$ = 47.

 $611 = 47 \cdot 13$.

The number-field sieve

Generalize $i \equiv i + N \pmod{N}$ $\rightarrow a \equiv a + bN \pmod{N}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly. For any *m* can find α so that factoring $m - \alpha$ produces factorization of N.

Optimal choice of log *m* is

$\rightarrow a - bm \equiv a - b\alpha \pmod{m - \alpha}$

 $(\mu + o(1))(\log N)^{2/3}(\log \log N)^{1/3}$.

mplete factorization of gruences $i \equiv 611 + i$ *i*'s.

 $= 2^{1}3^{0}5^{4}7^{1}$. $= 2^{6}3^{3}5^{2}7^{0}$. $= 2^{1}3^{1}5^{2}7^{3}$.

75 · 625 · 675 · 686 $^{8}7^{4} = (2^{4}3^{2}5^{4}7^{2})^{2}.$

 $14 \cdot 64 \cdot 75 - 2^4 3^2 5^4 7^2$

7 · 13.

The number-field sieve

Generalize $i \equiv i + N \pmod{N}$ $\rightarrow a \equiv a + bN \pmod{N}$ $\rightarrow a - bm \equiv a - b\alpha \pmod{m - \alpha}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly. For any *m* can find α so that factoring $m - \alpha$ produces factorization of N.

Optimal choice of log *m* is $(\mu + o(1))(\log N)^{2/3}(\log \log N)^{1/3}$.

RAM co

1993 Bu Smooth Sieve L^1 Find L^{0} .

with a –

Total RA

1993 Co

Total RA

using m

(Multipl don't se with AT ctorization of $\equiv 611 + i$

7¹. 7⁰. 7³.

 $575 \cdot 686$ $3^2 5^4 7^2)^2$.

 $75 - 2^4 3^2 5^4 7^2$

The number-field sieve

Generalize $i \equiv i + N \pmod{N}$ $\rightarrow a \equiv a + bN \pmod{N}$ $\rightarrow a - bm \equiv a - b\alpha \pmod{m - \alpha}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly.

For any *m* can find α so that factoring $m - \alpha$ produces factorization of *N*.

Optimal choice of log m is $(\mu + o(1))(\log N)^{2/3}(\log \log N)^{1/3}.$

RAM cost analysis

1993 Buhler-Lenst Smoothness bound Sieve $L^{1.923000}$ pa Find $L^{0.961500}$ pair with a - bm and aTotal RAM time *L* 1993 Coppersmith Total RAM time L using multiple nur (Multiple number don't seem to con with AT, factory,

The number-field sieve

Generalize $i \equiv i + N \pmod{N}$ $\rightarrow a \equiv a + bN \pmod{N}$ $\rightarrow a - bm \equiv a - b\alpha \pmod{m - \alpha}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly. For any *m* can find α so that factoring $m - \alpha$ produces factorization of N.

Optimal choice of log *m* is $(\mu + o(1))(\log N)^{2/3}(\log \log N)^{1/3}$.

1993 Buhler–Lenstra–Pomer Smoothness bound $L^{0.961500}$ Sieve $L^{1.923000}$ pairs (*a*, *b*). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ sm Total RAM time $L^{1.923000}$. 1993 Coppersmith:

5⁴7²}

ost analysis

Total RAM time $L^{1.901884}$ using multiple number fields

(Multiple number fields

with AT, factory, et al.)

don't seem to combine well

The number-field sieve

Generalize $i \equiv i + N \pmod{N}$ $\rightarrow a \equiv a + bN \pmod{N}$ $\rightarrow a - bm \equiv a - b\alpha \pmod{m - \alpha}$ for root $\alpha \in \mathbf{C}$ of nonzero integer poly. For any *m* can find α so that factoring $m - \alpha$ produces factorization of N.

Optimal choice of log *m* is $(\mu + o(1))(\log N)^{2/3}(\log \log N)^{1/3}$.

RAM cost analysis

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (*a*, *b*). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$. 1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields. (Multiple number fields don't seem to combine well

with AT, factory, et al.)

nber-field sieve

 $ze \ i \equiv i + N \pmod{N}$ $a + bN \pmod{N}$ $m \equiv a - b \alpha \pmod{m - \alpha}$ $\alpha\in \mathbf{C}$

ero integer poly.

m can find α

factoring $m - \alpha$

s factorization of N.

choice of log *m* is)) $(\log N)^{2/3}(\log \log N)^{1/3}$.

RAM cost analysis

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (*a*, *b*). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields) don't seem to combine well with AT, factory, et al.)

AT cost

Sieving in realist AT cost

sieve

 $N \pmod{N}$ mod N $\alpha \pmod{m-\alpha}$

poly.

 $d \alpha$

n-lpha

tion of N.

log m is $\frac{1}{3}(\log \log N)^{1/3}$.

RAM cost analysis

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (a, b). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields don't seem to combine well with *AT*, factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost method AT cost $L^{2.403750}$.

N)

 $m-\alpha$)

 $(N)^{1/3}$.

RAM cost analysis

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (*a*, *b*). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$. 1993 Coppersmith: Total RAM time $L^{1.901884}$

using multiple number fields.

(Multiple number fields don't seem to combine well with AT, factory, et al.)

AT cost analysis

 $AT \operatorname{cost} L^{2.403750}$.

Sieving is a disaster

in realistic cost metric.

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (a, b). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields don't seem to combine well with *AT*, factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \operatorname{cost} L^{2.403750}$.

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (*a*, *b*). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields) don't seem to combine well with AT, factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \cos L^{2.403750}$. Fix: find smooth using ECM. $AT \cos L^{1.923000}$

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (a, b). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields don't seem to combine well with *AT*, factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \operatorname{cost} L^{2.403750}$.

Fix: find smooth using ECM. AT cost $L^{1.923000}$.

Linear algebra is also a disaster. AT cost $L^{2.403750}$.

1993 Buhler–Lenstra–Pomerance: Smoothness bound $L^{0.961500}$. Sieve $L^{1.923000}$ pairs (a, b). Find $L^{0.961500}$ pairs with a - bm and $a - b\alpha$ smooth. Total RAM time $L^{1.923000}$.

1993 Coppersmith: Total RAM time $L^{1.901884}$ using multiple number fields.

(Multiple number fields don't seem to combine well with *AT*, factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \operatorname{cost} L^{2.403750}$.

Fix: find smooth using ECM. AT cost $L^{1.923000}$.

Linear algebra is also a disaster. AT cost $L^{2.403750}$.

Semi-fix: Reduce smoothness bounds to rebalance. $AT \operatorname{cost} L^{1.976052}$. (2001 Bernstein)

st analysis

hler–Lenstra–Pomerance: ness bound $L^{0.961500}$. ^{.923000} pairs (*a*, *b*). 961500 pairs

- bm and $a - b\alpha$ smooth. AM time $L^{1.923000}$.

ppersmith:

AM time $L^{1.901884}$

ultiple number fields.

e number fields em to combine well , factory, et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \cos L^{2.403750}$.

Fix: find smooth using ECM. $AT \text{ cost } L^{1.923000}$.

Linear algebra is also a disaster. $AT \cos L^{2.403750}$.

Semi-fix: Reduce smoothness bounds to rebalance. $AT \operatorname{cost} L^{1.976052}$. (2001 Bernstein)

The fact

1993 Co There ex that fact with san in RAM

Smooth Smaller

so need

Algorith such that Note: or

Algorith whether 2

tra-Pomerance: d $L^{0.961500}$.

irs (*a*, *b*).

S

a — *b*α smooth. 1.923000

: 1.901884

nber fields.

fields

nbine well

et al.)

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \operatorname{cost} L^{2.403750}$.

Fix: find smooth using ECM. AT cost $L^{1.923000}$.

Linear algebra is also a disaster. $AT \operatorname{cost} L^{2.403750}$.

Semi-fix: Reduce smoothness bounds to rebalance. $AT \operatorname{cost} L^{1.976052}$. (2001 Bernstein)

The factorization

1993 Coppersmith There *exists* an alg that factors any in with same #bits a in RAM time $L^{1.63}$ Smoothness bound Smaller than before so need more (*a*, *b*)

Algorithm *knows* a such that a - bmNote: one *m* work Algorithm uses EC whether $a - b\alpha_N$ ance:)

ooth.

AT cost analysis

Sieving is a disaster in realistic cost metric. $AT \operatorname{cost} L^{2.403750}$.

Fix: find smooth using ECM. $AT \cos L^{1.923000}$

Linear algebra is also a disaster. AT cost $L^{2.403750}$.

Semi-fix: Reduce smoothness bounds to rebalance. $AT \operatorname{cost} L^{1.976052}$. (2001 Bernstein)

The factorization factory

1993 Coppersmith:

- There *exists* an algorithm
- that factors any integer
- with same # bits as N
- in RAM time $L^{1.638587}$.
- Smoothness bound $L^{0.819290}$
- Smaller than before,
- so need more (*a*, *b*).
- Algorithm *knows* all (*a*, *b*)
- such that a bm is smooth
- Note: one *m* works for all Λ
- Algorithm uses ECM to che
- whether $a b\alpha_N$ is smooth

Sieving is a disaster in realistic cost metric. $AT \cos L^{2.403750}$.

Fix: find smooth using ECM. $AT \cos L^{1.923000}$

Linear algebra is also a disaster. $AT \cos L^{2.403750}$

Semi-fix: Reduce smoothness bounds to rebalance. $AT \operatorname{cost} L^{1.976052}$. (2001 Bernstein)

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same # bits as N in RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (a, b).

Algorithm *knows* all (*a*, *b*) such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

analysis

is a disaster cic cost metric. 1 2.403750

l smooth using ECM. L^{1.923000}.

lgebra is also a disaster. L^{2.403750}.

: Reduce smoothness

to rebalance.

 $L^{1.976052}$.

ernstein)

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same #bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (a, b).

Algorithm *knows* all (a, b)such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

Finding is slower Need to such tha RAM tir

er etric.

using ECM.

lso a disaster.

smoothness

ce.

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same #bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (a, b).

Algorithm *knows* all (a, b)such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

Finding this algori is slower than runn Need to precompusuch that a - bmRAM time $L^{2.0068}$

ster.

5S

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same # bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (*a*, *b*).

Algorithm *knows* all (*a*, *b*) such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

Finding this algorithm is slower than running it. Need to precompute all (a, l such that a - bm is smooth RAM time $L^{2.006853}$.

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same #bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (a, b).

Algorithm *knows* all (a, b)such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth. Finding this algorithm is slower than running it. Need to precompute all (a, b)such that a - bm is smooth. RAM time $L^{2.006853}$.

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same # bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (*a*, *b*).

Algorithm *knows* all (*a*, *b*) such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

Finding this algorithm is slower than running it. Need to precompute all (a, b) such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$ then precomputation cost becomes negligible.

The factorization factory

1993 Coppersmith: There *exists* an algorithm that factors any integer with same # bits as Nin RAM time $L^{1.638587}$.

Smoothness bound $L^{0.819290}$. Smaller than before, so need more (*a*, *b*).

Algorithm *knows* all (*a*, *b*) such that a - bm is smooth. Note: one *m* works for all *N*. Algorithm uses ECM to check whether $a - b\alpha_N$ is smooth.

Finding this algorithm is slower than running it. Need to precompute all (a, b) such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$ then precomputation cost becomes negligible.

The big problem: Coppersmith's algorithm has size $L^{1.638587}$. Huge AT cost; useless in reality.

corization factory

- ppersmith:
- *kists* an algorithm tors any integer
- ne #bits as N time $L^{1.638587}$.
- ness bound $L^{0.819290}$. than before, more (a, b).
- m *knows* all (*a*, *b*) It a - bm is smooth. ne *m* works for all *N*. m uses ECM to check $a - b \alpha_N$ is smooth.

Finding this algorithm is slower than running it. Need to precompute all (*a*, *b*) such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$, then precomputation cost becomes negligible.

The big problem: Coppersmith's algorithm has size $L^{1.638587}$. Huge AT cost; useless in reality.

Batch N Goal: O 1. Gene Test a – 2. Make close to When sr test eacl 3. After reorganiz relevant 4. Linea

factory

gorithm

iteger

s N

38587

d $L^{0.819290}$.

e,

)).

all (*a*, *b*)

is smooth.

is for all N.

CM to check

is smooth.

Finding this algorithm is slower than running it. Need to precompute all (a, b)such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$, then precomputation cost becomes negligible.

The big problem: Coppersmith's algorithm has size $L^{1.638587}$. Huge *AT* cost; useless in reality.

Batch NFS Goal: Optimize A 1. Generate (a, b)Test a - bm for si 2. Make many coj close to each (a, b When smooth a – test each $a - b\alpha_N$ 3. After all smoot reorganize: for each relevant (a, b) clos 4. Linear algebra.

Finding this algorithm is slower than running it. Need to precompute all (*a*, *b*) such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$ then precomputation cost becomes negligible.

The big problem: Coppersmith's algorithm has size $L^{1.638587}$. Huge AT cost; useless in reality.

ck

Batch NFS

- When smooth a bm is for
- test each $a b\alpha_N$ for smooth
- 3. After all smooths are fou
- reorganize: for each N, brin
- relevant (a, b) close togethe
- 4. Linear algebra.

Goal: Optimize AT asymptot

1. Generate (*a*, *b*) in paralle Test a - bm for smoothness

2. Make many copies of eac close to each (a, b) generated

Finding this algorithm is slower than running it. Need to precompute all (*a*, *b*) such that a - bm is smooth. RAM time $L^{2.006853}$.

Standard conversion of precomputation into batching: if there are enough targets, more than $L^{0.368266}$ then precomputation cost becomes negligible.

The big problem: Coppersmith's algorithm has size $L^{1.638587}$. Huge AT cost; useless in reality.

Batch NFS

Goal: Optimize AT asymptotics.

1. Generate (*a*, *b*) in parallel. Test a - bm for smoothness.

2. Make many copies of each $N_{\rm r}$ close to each (*a*, *b*) generator. When smooth a - bm is found,

3. After all smooths are found, reorganize: for each N, bring relevant (*a*, *b*) close together.

4. Linear algebra.

- test each $a b\alpha_N$ for smoothness.

- this algorithm
- than running it.
- precompute all (*a*, *b*)
- It a bm is smooth. ne $L^{2.006853}$.
- d conversion of utation into batching: are enough targets, an L^{0.368266},
- computation cost
- negligible.
- problem: Coppersmith's n has size $L^{1.638587}$.
- C cost; useless in reality.

Batch NFS

- Goal: Optimize AT asymptotics.
- 1. Generate (*a*, *b*) in parallel. Test a - bm for smoothness.
- 2. Make many copies of each N, close to each (*a*, *b*) generator. When smooth a - bm is found, test each $a - b\alpha_N$ for smoothness.
- 3. After all smooths are found, reorganize: for each N, bring relevant (*a*, *b*) close together.
- 4. Linear algebra.

Generate
$$(a, b)$$
.Is $a - bm$ smooth?If so, storeRepeat.Generate (a, b) .Is $a - bm$ smooth?If so, store.Repeat.Generate (a, b) .Is $a - bm$ smooth?If so, store.Repeat.Generate (a, b) .Is $a - bm$ smooth?If so, store.Repeat.Generate (a, b) .Is $a - bm$ smooth?If so, store.Repeat.If so, store.Repeat.

thm

ning it. Ite all (*a*, *b*) is smooth. 53

on of

to batching:

n targets, 56

_ _

on cost

2

Coppersmith's $L^{1.638587}$.

eless in reality.

Batch NFS

Goal: Optimize AT asymptotics.

1. Generate (a, b) in parallel. Test a - bm for smoothness.

2. Make many copies of each N, close to each (a, b) generator. When smooth a - bm is found, test each $a - b\alpha_N$ for smoothness.

3. After all smooths are found, reorganize: for each *N*, bring relevant (*a*, *b*) close together.

4. Linear algebra.

Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls <i>a</i> – <i>bm</i>	ls a bm
smooth?	smooth?
If so, store	If so, store.
Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
Is a \ne bm	Is a – bm
smøoth?	smooth?
If sq, store.	If so, store.
Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
ls a∕− bm	Is a – bm
smooth?	smooth?
If so, store.	If so, store.
Repeat.	Repeat.
Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls a – bm	Is a – bm
smooth?	smooth?
If so, store.	If so, store.
Repeat.	Repeat.

Batch NFS

Goal: Optimize AT asymptotics.

1. Generate (a, b) in parallel. Test a - bm for smoothness.

2. Make many copies of each N, close to each (a, b) generator. When smooth a - bm is found, test each $a - b\alpha_N$ for smoothness.

3. After all smooths are found, reorganize: for each *N*, bring relevant (*a*, *b*) close together.

4. Linear algebra.

Generate (a, b ls a – bm smooth? If so, store, Repeat. Generate/(a, l Is a $\neq bm$ smøoth? If sd, store. Repeat. Generate (a, l ls a∖– bm smo&th? If so, store. Repeat? Generate (a, l Is a – bm smooth? If so, store. Repeat.

ith's

5)

lg:

ality.

b).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
	ls a bm	ls a – bm
	smooth?	smooth?
	If so, store.	Nf so, store.
	Repeat.	Repeat.
b).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
	ls <i>a</i> – <i>bm</i>	Is $a + bm$
	smooth?	smooth?
	If so, store.	If so, store.
	Repeat.	Repeat.
b).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).
	ls <i>a</i> – bm	ls a – bm
	smooth?	smooth?
	If so, store.	If sø, store.
	Repeat.	Repeat.
b).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
	ls a – bm	ls a – bm
	smooth?	smooth?
	If so, store.	If so, store.
	Repeat.	Repeat.

Batch NFS

Goal: Optimize AT asymptotics.

1. Generate (a, b) in parallel. Test a - bm for smoothness.

2. Make many copies of each N, close to each (a, b) generator. When smooth a - bm is found, test each $a - b\alpha_N$ for smoothness.

3. After all smooths are found, reorganize: for each *N*, bring relevant (*a*, *b*) close together.

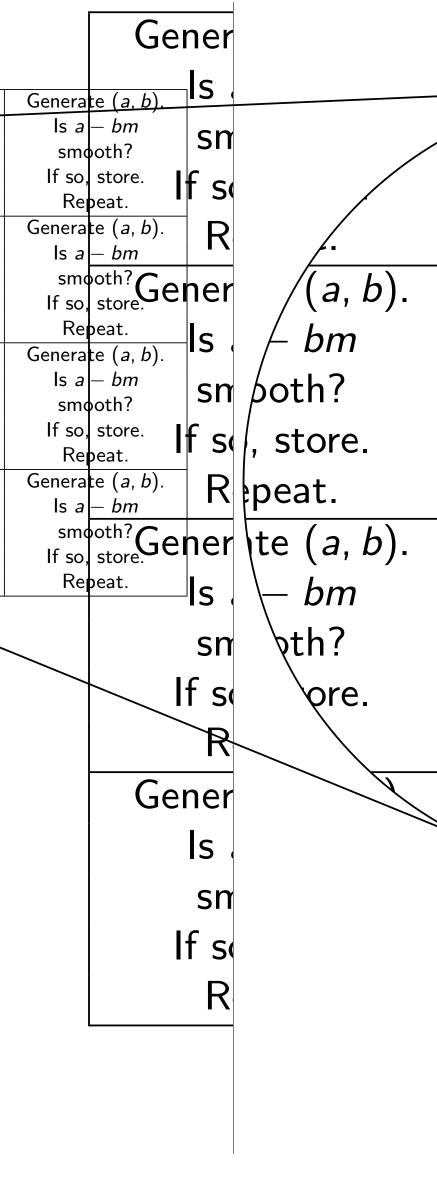
4. Linear algebra.

Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
ls a – bm	ls a bm	ls a – bm	ls a – bm
smooth?	smooth?	smooth?	smooth?
If so, store	If so, store.	If so, store.	If so, store.
Repeat.	Repeat.	Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
ls a 🗕 bm	ls <i>a</i> – <i>bm</i>	Is $a + bm$	ls <i>a</i> – <i>bm</i>
smøoth?	smooth?	smooth?	smooth?
If sq, store.	If so, store.	If so, store.	If so, store.
Repeat.	Repeat.	Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
ls a∖− bm	ls <i>a</i> – <i>bm</i>	ls a ┥bm	ls <i>a</i> – <i>bm</i>
smooth?	smooth?	smooth?	smooth?
If so, store.	If so, store.	If sø, store.	If so, store.
Repeat.	Repeat.	Repeat.	Repeat.
Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
ls a – bm	Is a – bm	ls a – bm	ls <i>a</i> – <i>bm</i>
smooth?	smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.	If so, store.
Repeat.	Repeat.	Repeat.	Repeat.
			·

FS

- ptimize AT asymptotics.
- rate (*a*, *b*) in parallel. - *bm* for smoothness.
- e many copies of each N, each (a, b) generator. mooth a - bm is found, h $a - b\alpha_N$ for smoothness.
- all smooths are found, ze: for each *N*, bring (*a*, *b*) close together.
- r algebra.

Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls <i>a</i> – <i>bm</i>	ls a bm	ls a – bm
smooth?	smooth?	smooth?
If so, store	If so, store.	∖f so, store.
Repeat.	Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls a 🗕 bm	ls a – bm	Is $a + bm$
smøoth?	smooth?	smooth?
If sq, store.	If so, store.	If so, store.
Repeat.	Repeat.	Repeat.
Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls a∕− bm	ls <i>a</i> – <i>bm</i>	ls a – bm
smooth?	smooth?	smooth?
If so, store.	If so, store.	If sø, store.
Repeat.	Repeat.	Repeat.
Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
ls a – bm	Is a – bm	ls a – bm
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.
Repeat.	Repeat.	Repeat.



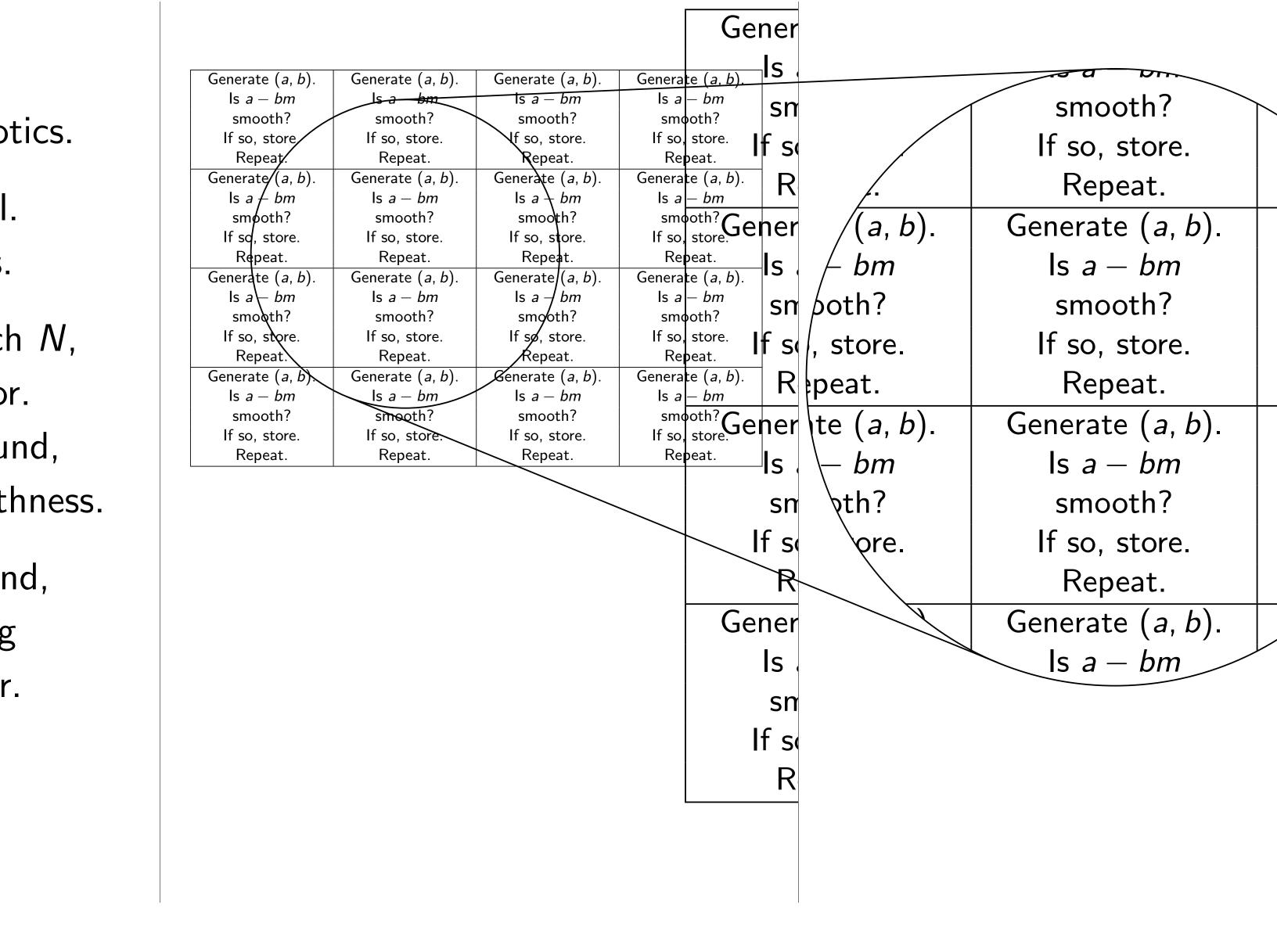
T asymptotics.

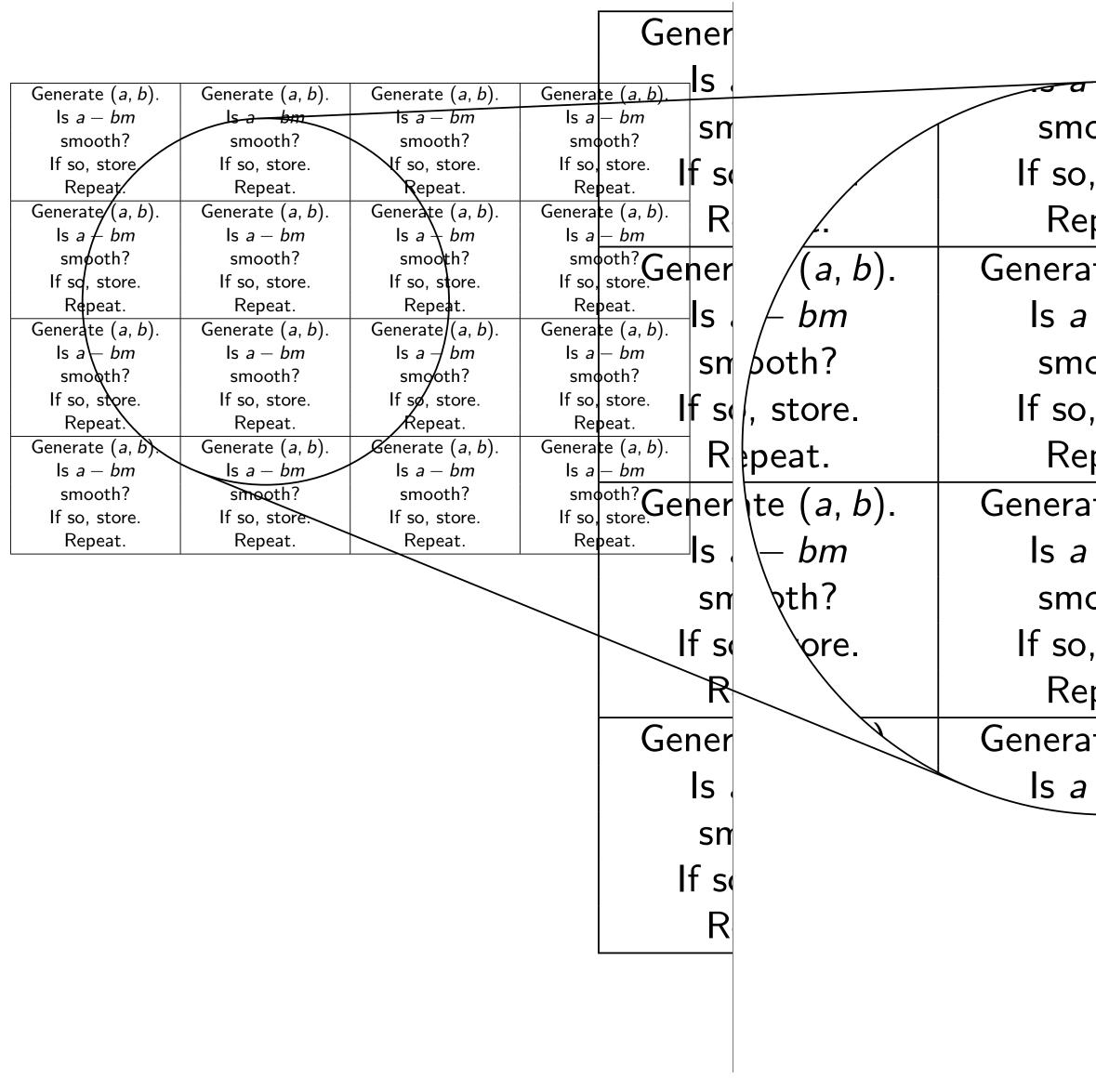
in parallel. noothness.

- bies of each N,
-) generator.
- *bm* is found,
- for smoothness.
- hs are found,
- ch N, bring
- se together.

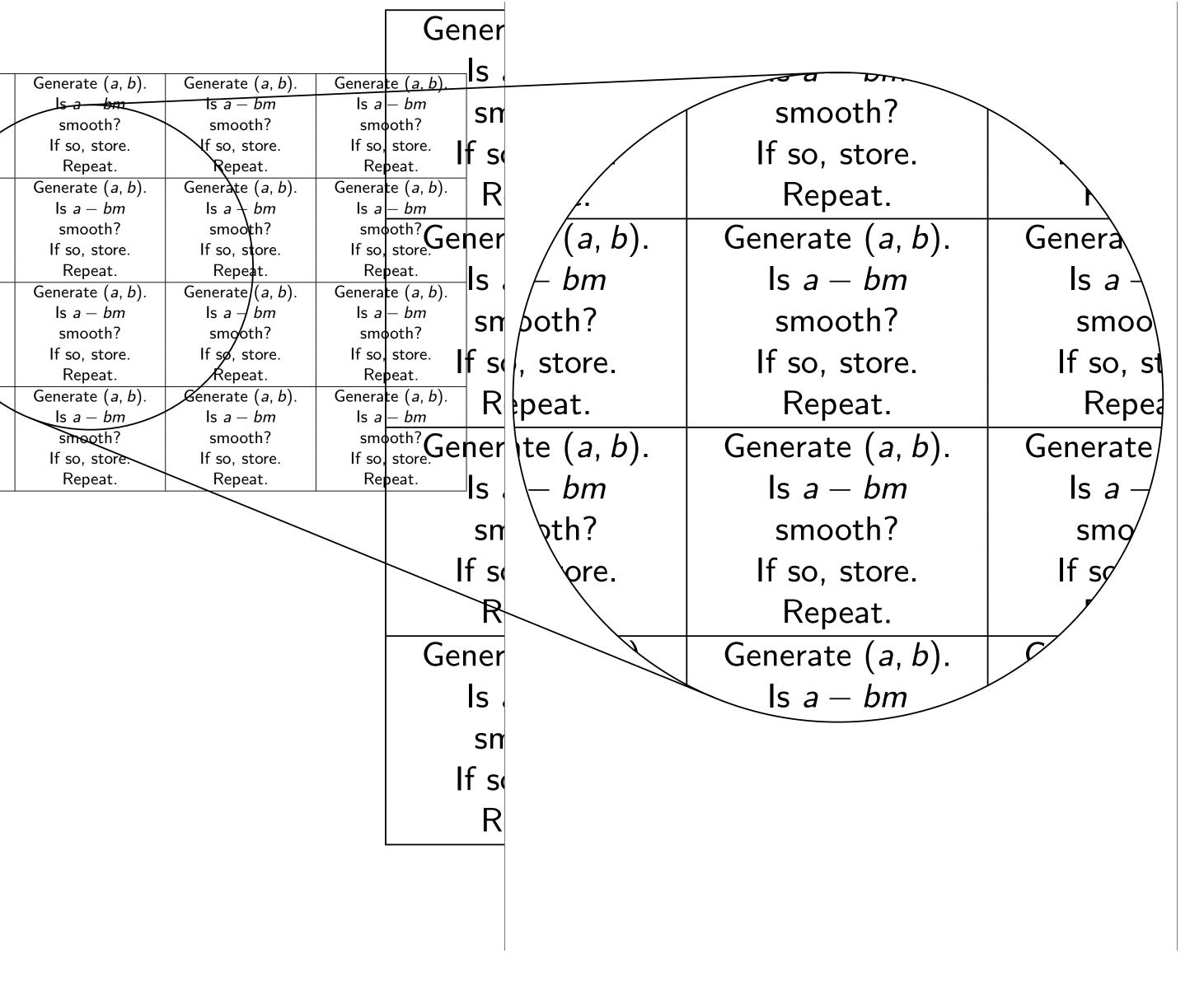
ner	Gei				
ls	te (<i>a, b</i>).	Conora	Generate (<i>a</i> , <i>b</i>).	Conorata (a, b)	$\int Concrate (a, b)$
	. ,			Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
sn	-bm		ls a - bm	sa bm	ls a – bm
	oth?		smooth?	smooth?	smooth?
f s	store.		If so, store.	If so, store.	If so, store
	beat.		Repeat.	Repeat.	Repeat.
R	te (<i>a</i> , <i>b</i>).		Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).
	— bm	ls a	Is $a + bm$	ls <i>a</i> – <i>bm</i>	ls a bm
hor	store.	smo	smooth?	smooth?	smøoth?
	store.	If so,	If so, store.	If so, store.	If so, store.
ls	beat.	Re	Repeat.	Repeat.	Repeat.
IS	te (<i>a</i> , <i>b</i>).	Genera	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).	Generate (a, b).
cn	– bm	ls a	ls a – bm	ls a – bm	Is a bm
sn	oth?	smo	smooth?	smooth?	smooth?
f s	store.	lf so,	If sø, store.	If so, store.	If so, store.
1 3	peat.	Re	Repeat.	Repeat.	Repeat.
R	te (<i>a</i> , <i>b</i>).	Genera	Generate (a, b).	Generate (a, b).	Generate (<i>a</i> , <i>b</i>).
	— bm	ls a	ls a – bm	Is a – bm	ls a – bm
	oth?	smo	smooth?	smooth?	smooth?
ner	store.	If so,	If so, store.	If so, store.	If so, store.
	peat.		Repeat.	Repeat.	Repeat.
12		•		1	
cn					
sn					
f s					
R					
ner	Ge				
ls					
-					

smo lf so, Rep (a, b). Generate ls *a* bm both? smo lf so, , store. peat. Rep te (*a*, *b*). Generate bm ls *a* oth? smo lf so, ore. Rep Generate ls *a* -S sn If so R

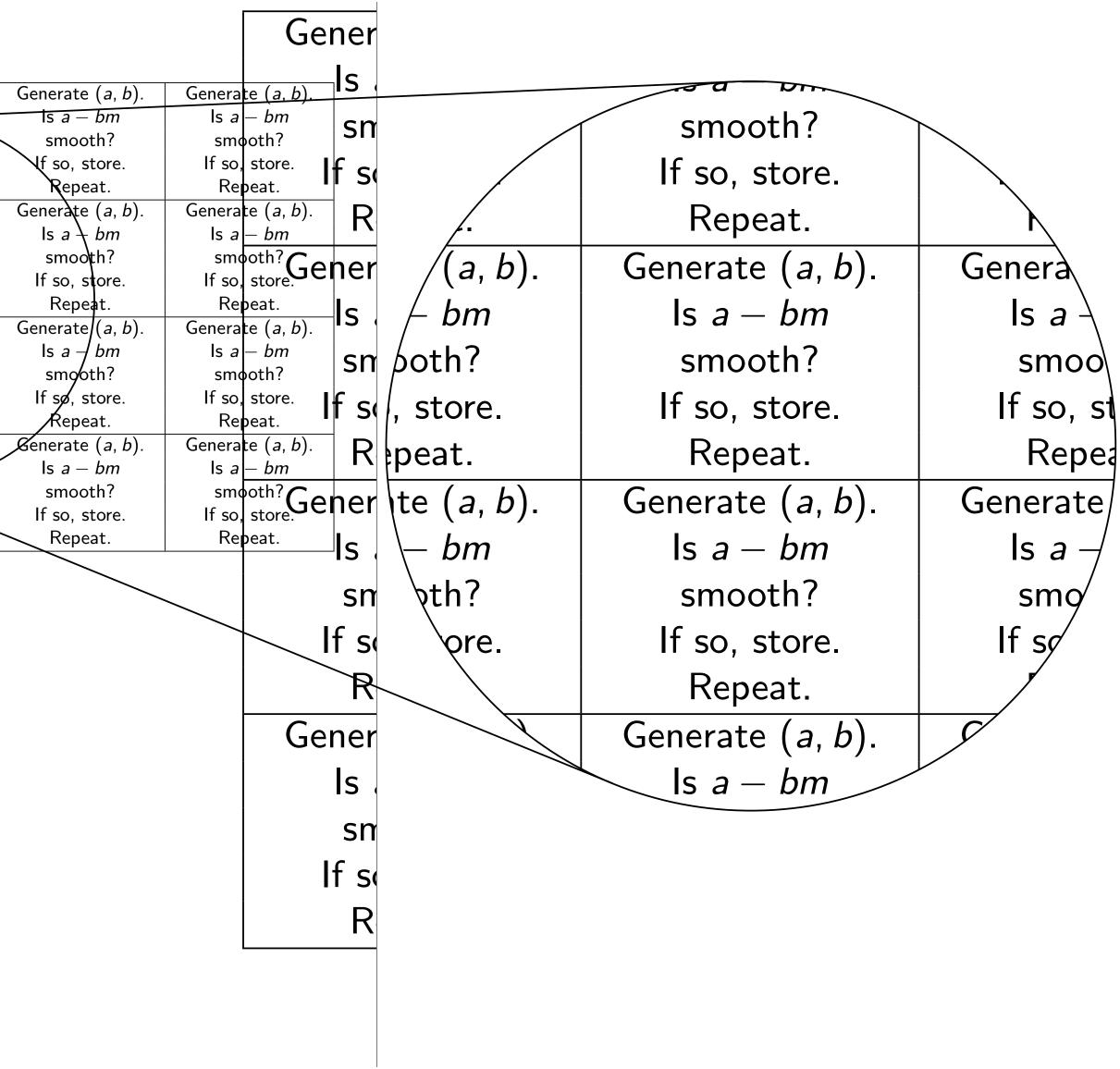




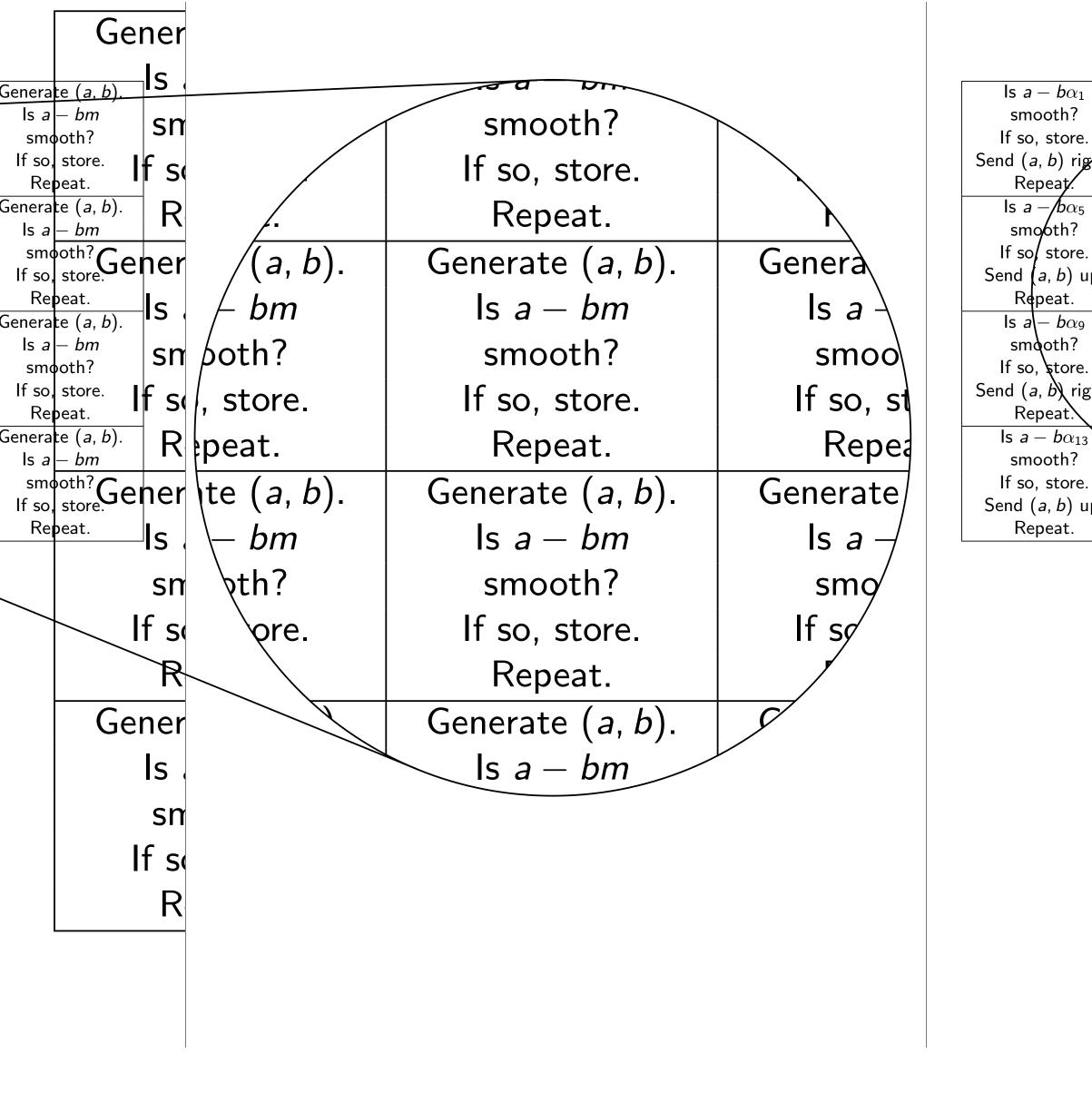
ooth?	
, store.	
epeat.	
ate (<i>a</i> , <i>b</i>).	Genera
-bm	Is $a \rightarrow$
ooth?	smoo
, store.	lf so, st
epeat.	Repea
ate (<i>a</i> , <i>b</i>).	Generate
-bm	ls <i>a</i> −∕
ooth?	smo/
, store.	lf sø
epeat.	
ate (<i>a</i> , <i>b</i>).	
– bm	



Is $a - b\alpha_1$ smooth? If so, store. Send (a, b) right Repeat/ Is a $-b\alpha_5$ smooth? If so, store. Send (*a*, *b*) up. Repeat. Is a $- \boldsymbol{b} lpha_{9}$ smooth? If so, store. Send (a, b) right Repeat? Is $a - b\alpha_{13}$ smooth? If so, store. Send (a, b) up. Repeat.



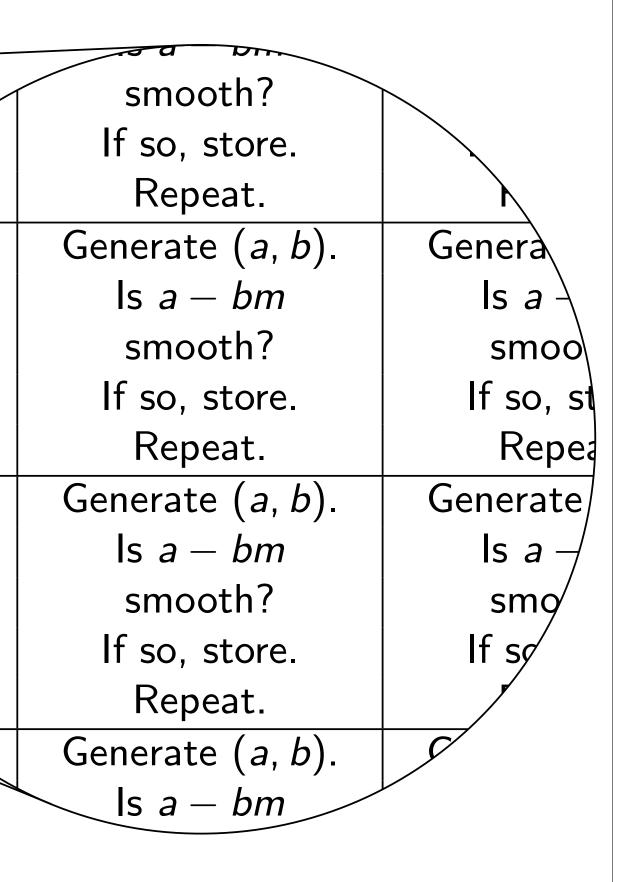
Is $a - b\alpha_1$	Is $a - b\alpha_2$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.
Repeat.	Repeat.
Is $a - b\alpha_5$	Is $a - b \alpha_6$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.
Repeat.	Repeat.
Is $a - b\alpha_9$	Is $a - b\alpha_{10}$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.
Repeat.	Repeat.
Is $a - b\alpha_{13}$	Is $a - b\alpha_{14}$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) teft.
Repeat.	Repeat.



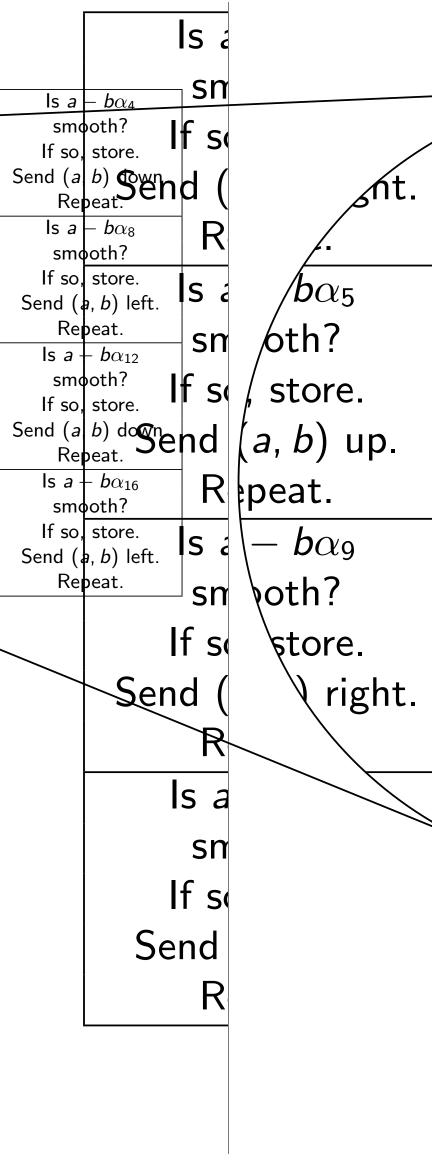
	Is $a - b \alpha_2$	Is $a - b \alpha_3$	
	smooth?	smooth?	
. /	If so, store.	If so, store.	
ght.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.	S
	Repeat.	Repeat.	
	Is $a - b \alpha_6$	Is $a - b\alpha_7$	
	smooth?	smooth?	
	If so, store.	If so, store.	
ıp.	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.	
	Repeat.	Repeat.	
	Is $a - b lpha_{10}$	Is $a - p\alpha_{11}$	
	smooth?	smooth?	
	If so, store.	If so,/store.	
ght.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.	S
	Repeat.	Repeat.	
\sim	Is $a - b lpha_{14}$	Is $a - b\alpha_{15}$	
	smooth?	smooth?	
	If so, store.	If so, store.	
ıp.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.	
	Repeat.	Repeat.	

	smooth?	
	If so, store.	
· ·	Repeat.	
(a, b).	Generate (<i>a</i> , <i>b</i>).	Genera
<i>⊢ bm</i>	ls <i>a</i> – <i>bm</i>	Is $a \rightarrow$
∮oth?	smooth?	smoo
, store.	If so, store.	If so, st
epeat.	Repeat.	Repea
te (<i>a</i> , <i>b</i>).	Generate (<i>a</i> , <i>b</i>).	Generate
igarrow bm	ls <i>a</i> – <i>bm</i>	Is <i>a</i> –∕
vth?	smooth?	smo/
ore.	If so, store.	If so
	Repeat.	
	Generate (<i>a</i> , <i>b</i>).	9
	Is a – bm	

Is $a - b\alpha_1$ smooth? If so, store. Send (a, b) right. Send (a, b) right. Repeat. Is $a - b\alpha_5$ smooth? If so store. Send (a, b) up. Repeat.	Is $a - b\alpha_2$ smooth? If so, store. Send (a, b) right. Repeat. Is $a - b\alpha_6$ smooth? If so, store. Send (a, b) left.	Is $a - b\alpha_3$ smooth? If so, store. Send (a, b) right. Repeat. Is $a - b\alpha_7$ smooth? If so, store.	Is $a - b\alpha_4$ smooth? If so, store. Send (a, b) down. Repeat. Is $a - b\alpha_8$ smooth? If so, store.
Send (a, b) right.Send (a, b) right.Repeat.Is $a - b\alpha_5$ smooth?If so store.Send (a, b) up.	Send (a, b) right. Repeat. Is $a - b \alpha_6$ smooth? If so, store.	Send (a, b) right. Repeat. Is $a - b\alpha_7$ smooth? If so, store.	Send (a, b) down. Repeat. Is $a - b\alpha_8$ smooth?
Repeat.Is $a - b\alpha_5$ smooth?If so, store.Send (a, b) up.	Repeat. Is $a - b\alpha_6$ smooth? If so, store.	Repeat. Is $a - b\alpha_7$ smooth? If so, store.	Repeat. Is $a - b\alpha_8$ smooth?
Is $a - b\alpha_5$ smooth? If so, store. Send (a, b) up.	Is $a - b \alpha_6$ smooth? If so, store.	Is $a - b\alpha_7$ smooth? If so, store.	Is $a - b\alpha_8$ smooth?
smooth? If so/store. Send (<i>a</i> , <i>b</i>) up.	smooth? If so, store.	smooth? If so, store.	smooth?
If so store. Send (a, b) up.	If so, store.	lf so, store.	
Send (a, b) up.	, , , , , , , , , , , , , , , , , , ,		If so, store.
	Send (<i>a</i> , <i>b</i>) left.		
Repeat.		Send (<i>a</i> , þ) left.	Send (<i>a</i> , <i>b</i>) left.
	Repeat.	Repeat.	Repeat.
Is $a - b\alpha_9$	Is $a - b\alpha_{10}$	Is $a - p \alpha_{11}$	Is $a - b\alpha_{12}$
smooth?	smooth?	smooth?	smooth?
If so, store.	If so, store.	If so,/store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.	Send (<i>a</i> , <i>b</i>) down.
Repeat.	Repeat.	Repeat.	Repeat.
Is $a - b\alpha_{13}$	Is $a - b \alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b \alpha_{16}$
smooth?	smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
Repeat.	Repeat.	Repeat.	Repeat.

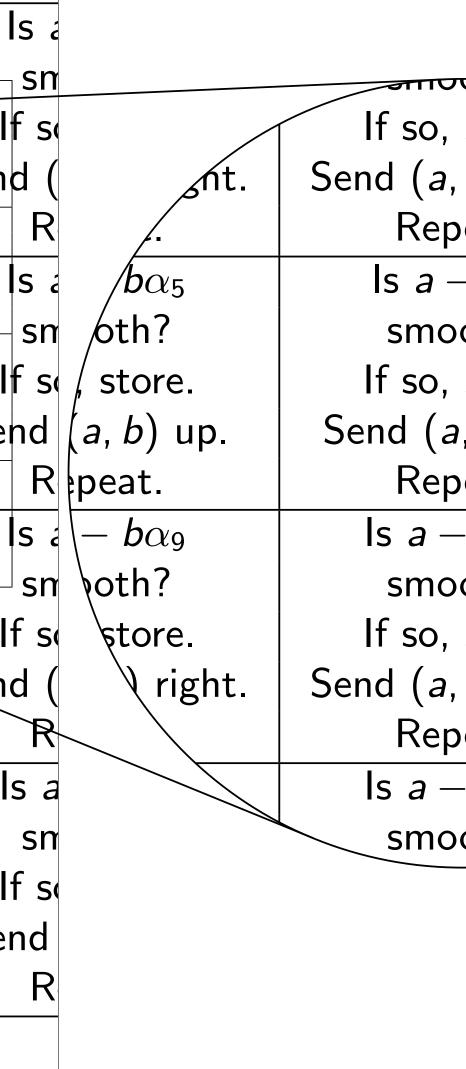


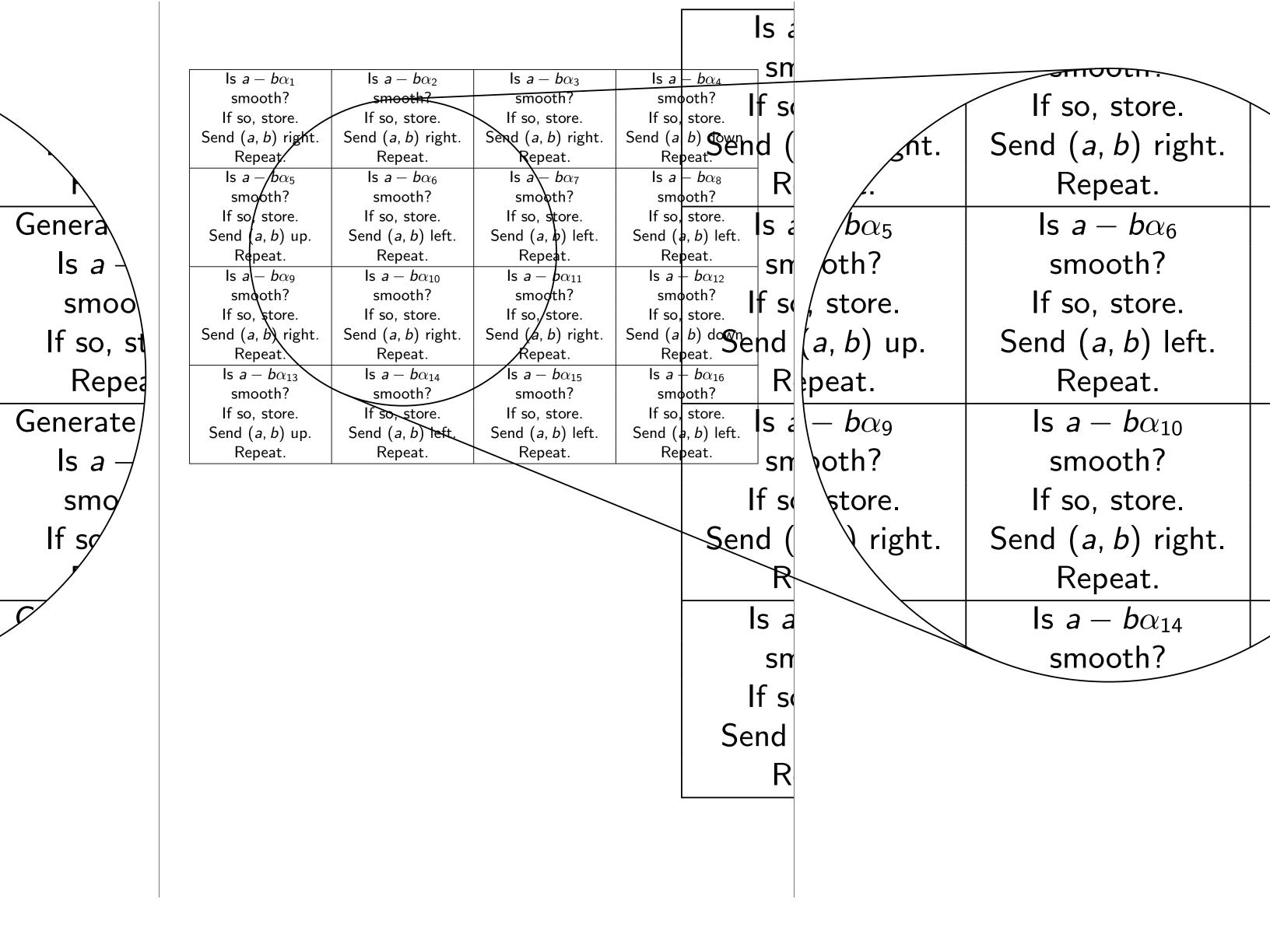
Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b \alpha_3$
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.
Repeat.	Repeat.	Repeat.
Is $a - b\alpha_5$	Is $a - b \alpha_6$	Is $a - b\alpha_7$
smooth?	smooth?	smooth?
If so store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , b) left.
Repeat.	Repeat.	Repeat.
Is $a - b\alpha_9$	Is $a - b\alpha_{10}$	Is $a - p \alpha_{11}$
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so,/store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.
Repeat.	Repeat.	Repeat.
Is $a - b\alpha_{13}$	Is $a - b \alpha_{14}$	$ls a - b\alpha_{15}$
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
Repeat.	Repeat.	Repeat.

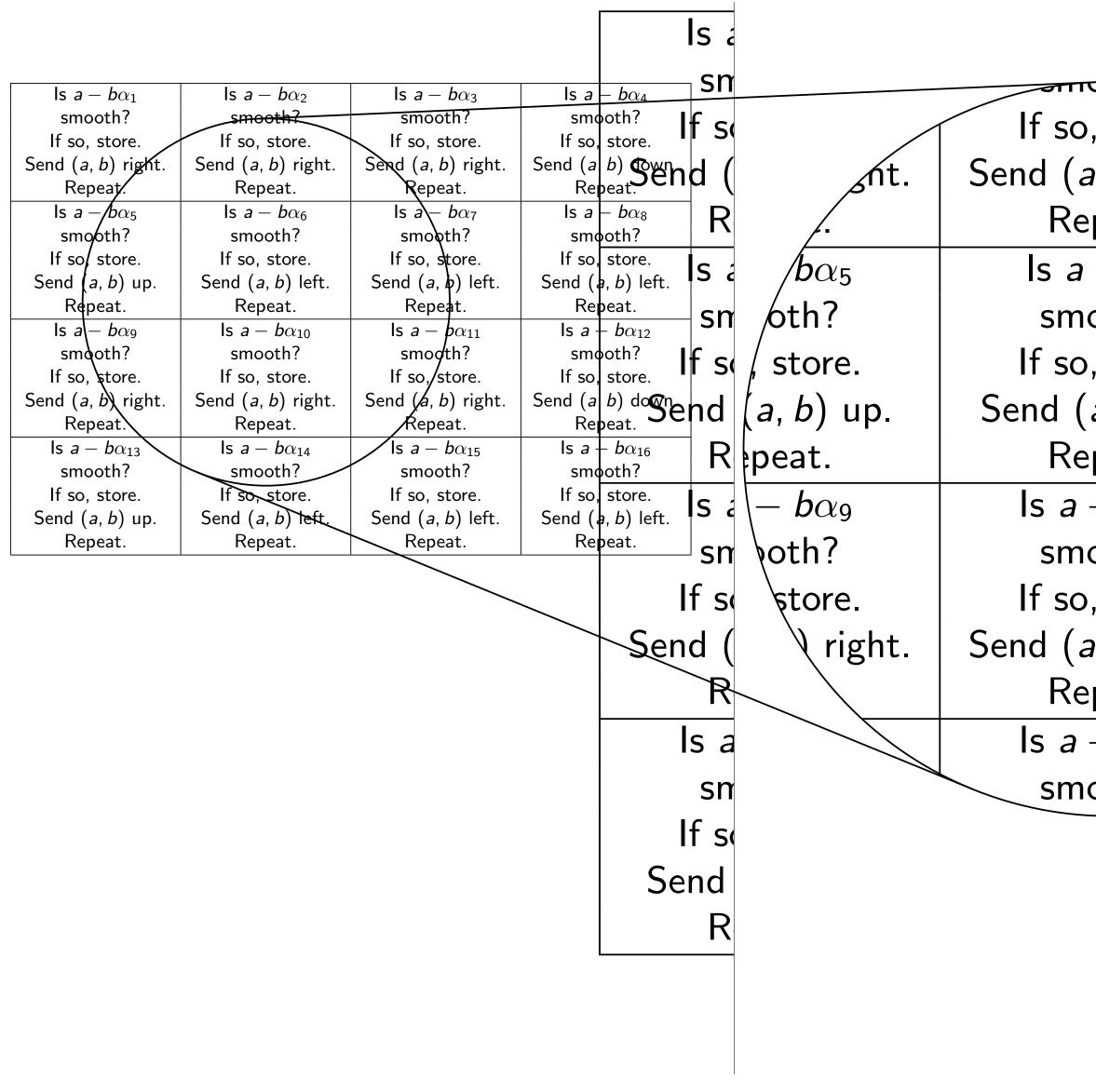


oth?	
store.	
eat.	
e (<i>a, b</i>).	Genera
- bm	$ $ Is $a \rightarrow $
oth?	smoo
store.	If so, st
eat.	Repea
e (<i>a</i> , <i>b</i>).	Generate
- bm	Is $a - /$
oth?	smo/
store.	If so
eat.	
e (<i>a, b</i>).	
- bm	

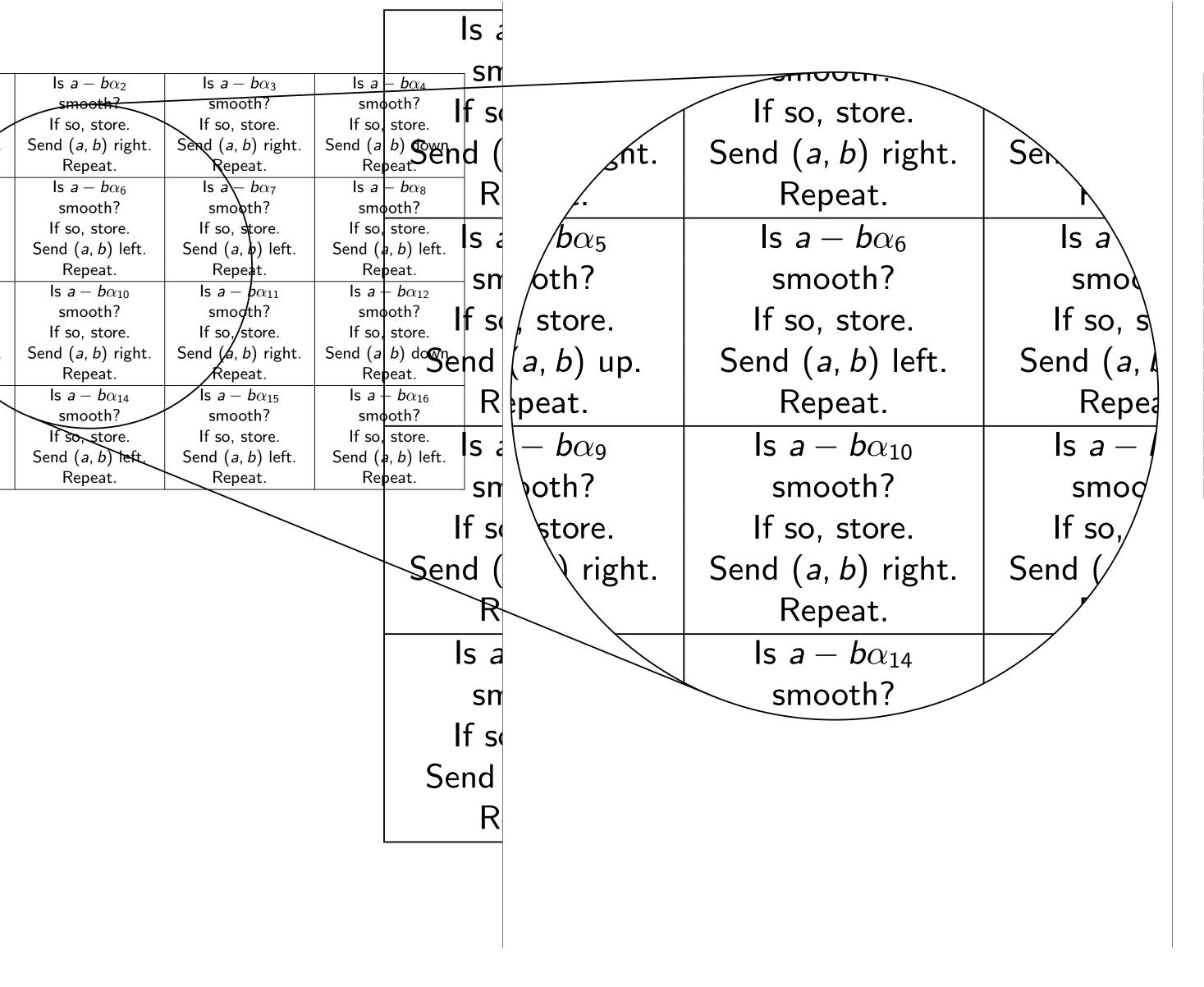
Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	ls a	$- \boldsymbol{b} lpha_4$	_] Sr
smooth?	smooth?	smooth?	smo	ooth?	fc
If so, store.	If so, store.	If so, store.	lf so,	store.	1 3
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.	Send (<i>a</i>	^{b)} Sen	Ы
Repeat.	Repeat.	Repeat.	Re	beat.	μ
Is $a - b\alpha_5$	Is $\mathit{a} - \mathit{b} lpha_{6}$	Is a $b\alpha_7$	ls a	$- b lpha_8$	R
smooth?	smooth?	smooth?	smo	oth?	
If so, store.	If so, store.	lf so, store.	lf so,	store.	s
Send <i>(a, b)</i> up.	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.	Send (a, b) left.	5
Repeat.	Repeat.	Repeat.	Re	peat.	sr
Is $a - b\alpha_9$	Is $\textit{a} - \textit{b} lpha_{10}$	Is $a - p lpha_{11}$	ls <i>a</i> -	– $m{b}lpha_{12}$	5
smooth?	smooth?	smooth?	smo	ooth?	f s
If so, store.	If so, store.	If so,/store.	lf so,	store.	
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.	Send (<i>a</i>	b) do g n	nd
Repeat.	Repeat.	Repeat.	Re	beat. JC	
Is $a - b\alpha_{13}$	Is $\pmb{a} - \pmb{b} lpha_{14}$	Is $a - b\alpha_{15}$	ls <i>a</i> -	– $m{b}lpha_{16}$	R
smooth?	smooth?	smooth?		oth?	
If so, store.	If so, store.	If so, store.	_	store.	ls
Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.	•	a, b) left.	
Repeat.	Repeat.	Repeat.	Re	peat.	[]] sr
					51
					f s
					.1
				Sen	a
					<u> </u>
					П
					Sä
				1	50
					sr
				1	ſ,
				I	f s
				Se	nd
					F
					R



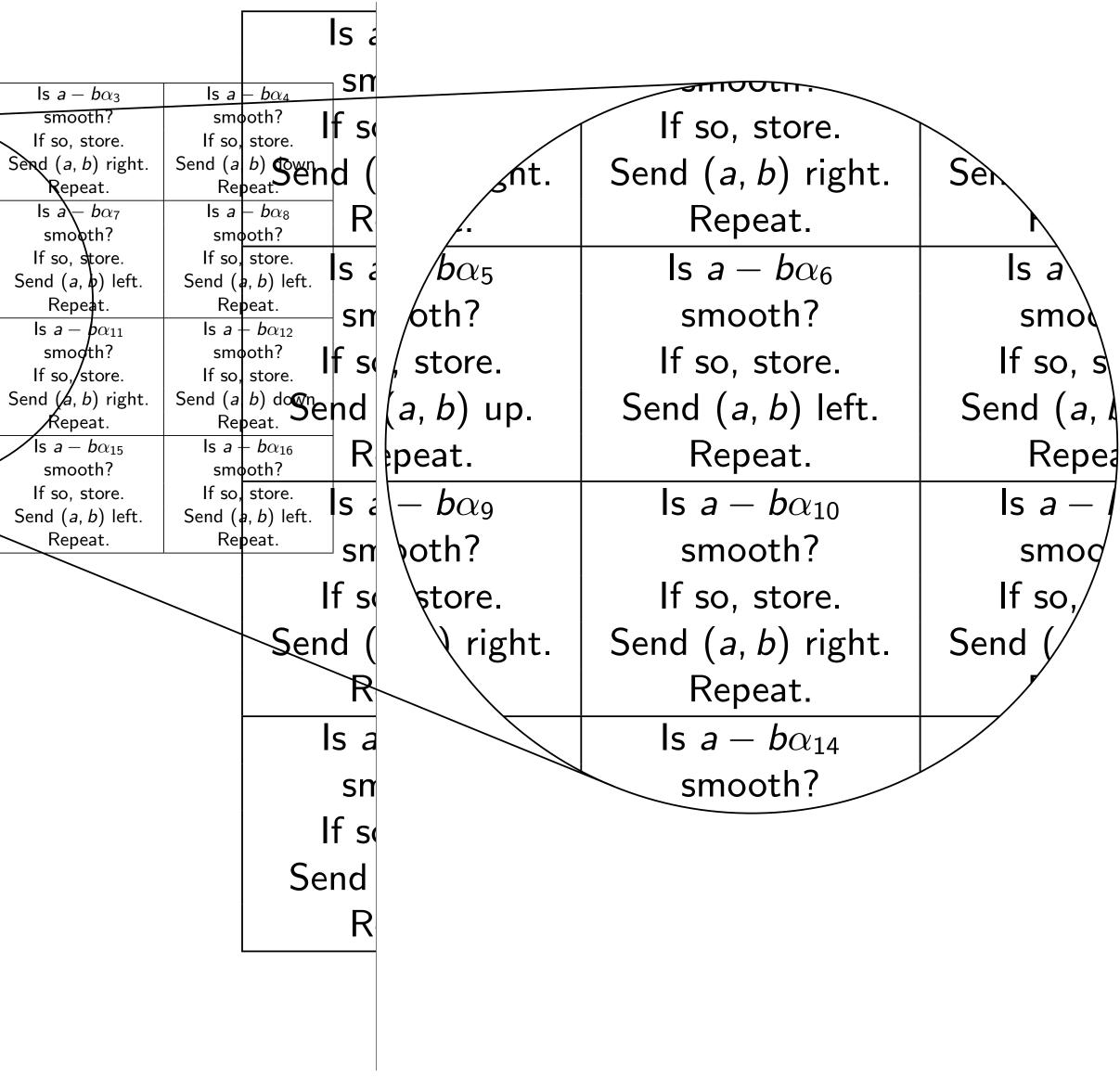




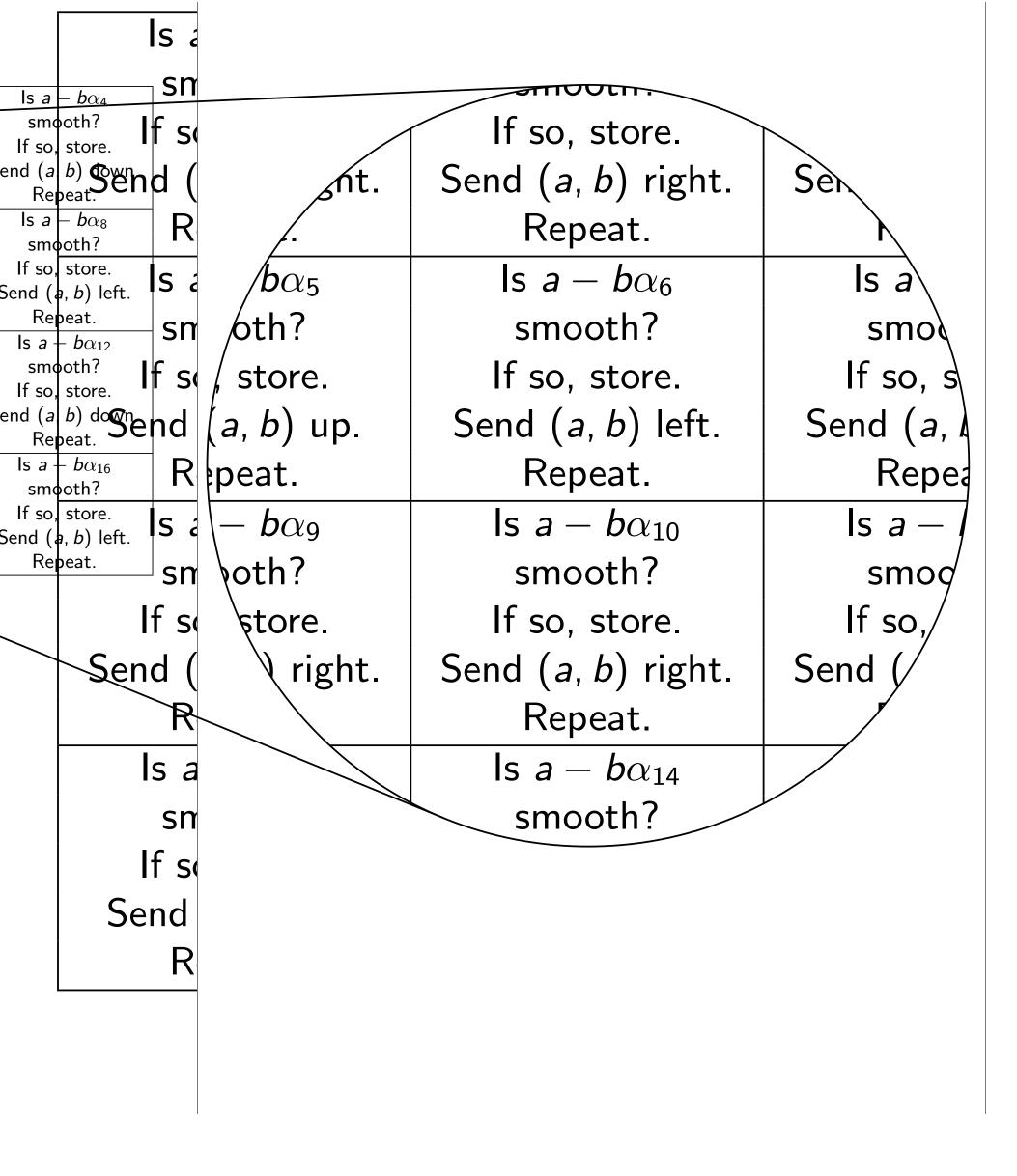
, store.	
a, b) right.	Ser
epeat.	
$- \boldsymbol{b} lpha_{6}$	ls a
ooth?	smod
, store.	lf so, s
<i>a</i> , <i>b</i>) left.	Send (a, k
epeat.	Repea
$- \boldsymbol{b} lpha_{10}$	ls <i>a</i> – /
ooth?	smod
, store.	lf so,/
a, b) right.	Send (/
epeat.	
$- \boldsymbol{b} lpha_{14}$	
ooth?	



_					
	$ls a - b\alpha_1$	$ \mathbf{s} _{\theta} = b\alpha_2$	ls a – bo ₃	Is $a - b\alpha_4$	
	smooth?	smooth?	smooth?	smooth? If so, store.	
	If so, store. Send (a, b) right. Repeat.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	Send (a, b) down. Repeat.	Send
	ls a - bcs	Repeat. Is a - bos	Repeat. Is a - bor	ls a - bca	
	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	If
	Send (a, b) up.				Ser
1	Repeat. Is a - boy smooth?	Repeat. Is a - boso smooth?	Repeat. Is a - bo ₁₁ smooth?	Repeat. Is a - bo ₁₂ smooth?	
	If so, store.	If so, store.	If so, store,	If so, store. Send (a. b) down.	If
	Send (a, b) right. Repeat. Is a - bo ₁₃	Send (a, b) right. Repeat. Is a - boss	Send (a, b) right. Repeat.	Repeat.	
i i			Repeat. Is a - brizs smooth?		ls is
	If so, store. Send (a, b) up.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If Sen
	Repeat.	Repeat.	Repeat.	Repeat.	
	ls a – bas smooth?	ls a – ba ₂ smooth?	Is a - bog smooth?	Is a - bos smooth?	
	If so store	If so store	If so, store	If so, store. Send (a, b) down.	
	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	Send
	Is a - bos smooth?	Is a - bos smooth?	Is a - bor smooth?	Is a - box smooth?	
	If so, store.	If so, store. Send (a, b) left.	If so, store.	If so, store.	If Ser
	Repeat.	Repeat. Is a - brizo	Send (a, b) left. Repeat. Is a - bo ₁₁	Send (a, b) left. Repeat. Is a - bo ₁₂	
	smooth?				
	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.	If Send
	Is a - borr	Repeat.	Repeat. Is a - boss	Repeat. Is a - bo ₁₆	
	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	
	If so, store. Send (a, b) up. Repeat.	If so, store. Send (a, b) left. Repeat.	If so, store. Send (a, b) left. Repeat.	If so, store. Send (a, b) left. Repeat.	
	Repeat.	Repeat.	Repear.	Repeat.	
	Is a - bas smooth?	Is a - bo ₂ smooth?	Is a - bog smooth?	Is a - box smooth?	
	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.	If Send
i -	Repeat.	Repeat. Is a - bos	Repeat.	Repeat.	i E
	smooth?	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	
	If so, store. Send (a, b) up.				Ser
i .	Repeat. Is a - boy	Repeat. Is a - boso	Repeat. Is a - bo ₁₁	Repeat. Is a - bo ₁₂	i 🗖
	smooth? If so, store. Send (a, b) right.	smooth? If so, store,	smooth? If so, store. Send (a, b) right.	smooth? If so, store. Send (a, b) down.	
	Send (a, b) right. Repeat.	If so, store. Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	Send
	Is a - bo ₁₃ smooth?	Is a - bost smooth?	ls a - boss smooth?	ls a - boss smooth?	Is
	If so, store,	If so, store,	If so, store. Send (a, b) left.	If so, store,	If
	Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Repeat.	Send (a, b) left. Repeat.	Ser
⊢					
	$ls a - b\alpha_1$	Is $a - b\alpha_2$	ls a – bog	Is $a - b\alpha_4$	
	smooth? If so, store.	smooth? If so, store.	smooth?	smooth? If so, store.	16
	Send (a, b) right.		If so, store. Send (a, b) right.	Send (a, b) down.	Send
	Repeat. Is a - bos smooth?	Repeat. Is a - bos smooth?	Repeat. Is a - boy smooth?	Repeat. Is a - bog smooth?	
	If so, store.	smooth? If so, store,	if so, store,	If so, store,	If
	Send (a, b) up. Repeat. Is a - boy	Send (a, b) left. Repeat. Is a - boso	Send (a, b) left. Repeat. Is a - ba ₁₁	Send (a, b) left. Repeat. Is a - bo ₁₂	Ser
	smooth?	smooth?	smooth?	smooth?	i
	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.	If Send
	Repeat.	Repeat. Is a - bo ₁₄	Repeat. Is a - bo ₁₅	Repeat. Is a - bo ₁₆	- Is
i i	smooth?	smooth?	smooth?	smooth?	
	If so, store. Send (a, b) up.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If Ser
	Repeat.	Repeat.	Repeat.	Repeat.	
					-
	Is a - bas smooth?	Is $a - b\alpha_2$ smooth?	ls a – bog smooth?	Is a - bos smooth?	
	If so, store.	If so, store.	If so, store.	If so, store.	If
i	Send (a, b) right. Repeat. Is a - bros	Send (a, b) right. Repeat. Is a - bos	Repeat.	Send (a, b) down. Repeat. Is a - box	Send
	smooth?	smooth?	smooth?	smooth?	
	If so, store. Send (a, b) up.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If so, store. Send (a, b) left.	If Ser
	Repeat.	Repeat.	Repeat.	Repeat.	1
	smooth? If so, store.	smooth?	smooth?	smooth?	
	Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (<i>a</i> , <i>b</i>) right.	If so, store. Send (a, b) down.	If Send
	Repeat. Is a - bo ₁₃	Repeat. Is a - boss	Repeat. Is a - bass	Repeat. Is a - bo ₁₆	b b
	smooth7		smooth?	smooth? If so, store,	
	If so, store. Send (a, b) up. Repeat.	If so, store. Send (a, b) left. Repeat.	If so, store. Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Ser



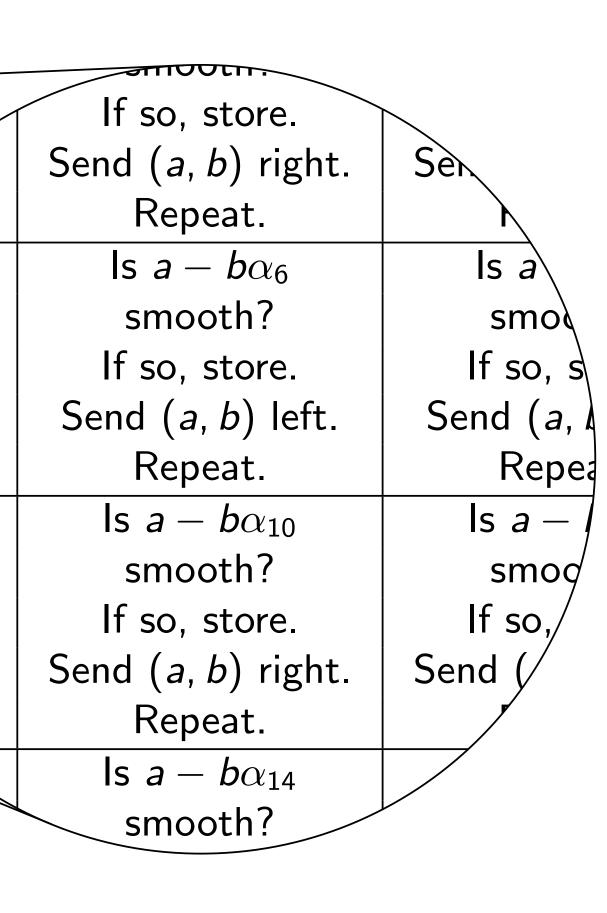
				-					- 1		
Is $a - b\alpha_1$	$\ln a - b\alpha_2$	$\ln a - b\alpha_3$	Is $a - b\alpha_4$	L	Is $a - b\alpha_1$	ls a – bo ₂	$ls a - b\alpha_1$	$\ln a - b\alpha_4$		Is $a - b\alpha_1$	Is $a - b\alpha_2$
smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.
Send (a, b) right. Repeat.	Send (a h) right	Scard (a, b) right			Send (a, h) right		Send (a h) right	Send (a, b) down.			
ls a - hus	Repeat. Is a - bos	Repeat. Is a - bor	Repeat. Is a - box		Repeat. Is a - bos	Repeat. Is a - bos	Repeat. Is a - bor	Is a - box		Repeat. Is a - bos	Repeat. Is a - bos
smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store.	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,
Send (a, b) up.	Send (a, b) left.	Send (a, b) left. Repeat.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.
Repeat. Is a - boy	Repeat. Is a - bo ₁₀	$ls a - b\alpha_{11}$	Repeat. Is a - bo ₁₂	i I	Repeat. Is a - bay	Repeat. Is a - ba ₁₀	Repeat. Is a - bo ₁₁	Repeat. Is a - bo ₁₂	l i	Repeat. Is a - boy	Repeat. Is a - bo ₁₀
smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.
Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.		Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Reneat		Send (a, b) right.	Send (a, b) right.
Is a - bo ₁₃ smooth?	Is a - bass smooth?	Is $a - b\alpha_{15}$ smooth?	ls a - bo ₁₆ smooth?	11	Is a - bass smooth?	Is $a = b\alpha_{14}$ smooth?	$b a - b \alpha_{15}$	smooth?	-	Is $a - b\alpha_{11}$ smooth?	ls a – bα ₁₄ smooth?
If so, store.	If so, store.	If so, store.	If so, store.		If so, store.	so, store,	If so, so	If so, store.		If so, store.	If so, store.
Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up, Repeat	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.
				<u> </u>					<u> </u>		
$ s a - b\alpha_1$	k a – hrs	$ \mathbf{s} _{\mathbf{z}} = b_{00}$	ls a - hru	Ι.	h a - hru	is a - box	ls a - hou			$\ln a - b\alpha_1$	ls a - hou
smooth?	smooth?	smooth?	smooth?		smooth?	smooth?	smooth?	smooth?		smooth?	smooth?
If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.		If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.		If so, store. Send (a, b) right.	If so, store. Send (a, b) right
Repeat. Is a - bos	Repeat. Is a - bos	Repeat. Is a - bor	Repeat. Is a - box	1/	Repeat. Is a - bos	Repeat. Is a - bos	Repeat. Is a - bor	Repeat.	ΛI	Repeat. Is a - bos	Repeat. Is a - bos
smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	11	smooth? If so, store.	smooth? If so, store,
Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.
Repeat. Is a - boy	Repeat. Is a - bo ₁₀	Repeat. Is a - bo ₁₁	Repeat. Is a - bo ₁₂	- A - I	Repeat. Is a - boy	Repeat. Is a - bo ₁₀	Repeat. Is a - bo ₁₁	Repeat. Is a - boys	11	Repeat. Is a - bog	Repeat. Is a - ba ₁₀
smooth? If so, store,	smooth? If so, store.	smooth? If so, store,	smooth? If so, store,	N	smooth? If so, store,	smooth? If so, store,	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store,
Send (a, b) right.	Send (a, b) right.	Send (a, b) right.	Send (a, b) down.	$ \mathbf{N} $	Send (a, b) right.	Send (a, b) right.	Send (a, b) right.	Send (a, b) down.	1	Send (a, b) right.	Send (a, b) right
Repeat. Is a - bo ₁₃	Repeat. Is a - bass	Repeat. Is a - bo ₁₅	Repeat. Is a - bo ₁₆	IN	Repeat. Is a - ba ₁₃	Repeat. Is a - ba ₁₄	Repeat. Is a - bo ₁₅	Repeat. Is a - bo ₁₆	1	Repeat. Is $a - b\alpha_{11}$	Repeat. Is a - bo ₁₄
smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store		smooth? If so, store.	smooth? If so, store.
Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a. b) left.	Send (a, b) left.		Stud (a, b) up.		Send (a, b) left.	Send (a, b) seft.		Send (a, b) up.	
Repeat.	Kepeat.	Repeat.	Repeat.	1	upeat.	Repeat.	Repeat.	R	1	Repeat.	Repeat.
				1				/	1		
Is a - bas smooth?	Is $a - b\alpha_2$ smooth?	Is a - bog smooth?	Is $a - b\alpha_4$ smooth?		Is $a - b\alpha_1$ smooth?	$b a - b \alpha_2$	Is a	Is $a - b\alpha_4$ smooth?		Is $a - b\alpha_1$ smooth?	Is a - boy smooth?
If so, store.	If so, store.	If so, store.	If so, store.		If so, store.	If so, sore.	If so, store.	If so, store.		If so, store.	If so, store.
Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	11	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	11	Send (a, b) right. Repeat.	Send (a, b) right Repeat.
Is a - bos smooth?	Is $a - b\alpha_6$ smooth?	Is a - boy smooth?	Is a - box smooth?		Is $a - b\alpha_5$ smooth?	Is a - bos smooth?	Is a - boy shouth?	ls a – bog smooth?		Is a - bos smooth?	Is a - box smooth?
If so, store.	If so, store,	If so, store,	If so, store,		If so, store.	If so, store,	If so, some.	If so, store.		If so, store,	If so, store,
Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.
Is a - boy smooth?	Is a - boxed smooth?	Is $a = b\alpha_{11}$ smooth?	Is a - bo ₁₂ smooth?	11	ls a – boy smooth?	Is $a = b\alpha_{10}$ smooth?	Is a - bo ₁₁ smooth?	$a = b\alpha_{12}$ srb eth?	11	Is a - bay smooth?	Is a – bα ₁₀ smooth?
If so, store.	If so, store.	If so, store.	If so, store.		If so, store.	If so, store,	If so, store.	If so, store		If so, store.	If so, store.
Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.		Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down Repeat.	N	Send (a, b) right. Repeat.	Send (a, b) right Repeat.
Is a - bo ₁₃ smooth?	Is a - base smooth?	Is a - bo ₁₅ smooth?	Is a - bo ₁₆ smooth?	11	Is a - bass smooth?	Is a - bost smooth?	Is a - bo ₁₅ smooth?	Is a - boys smooth?		Is $a = b\alpha_{11}$ smooth?	Is a - bo ₁₄ smooth?
If so, store. Send (a, b) up.	If so, store. Send (a, h) left	If so, store. Send (a, b) left	If so, store. Send (a, b) left		If so, store. Send (a, b) up.	If so, store. Send (a h) left	If so, store. Send (a, b) left	If so, store. Send (a, b) left		If so, store. Senada, b) up.	If so, store. Send (a h) left
Repeat.	Repeat.	Repeat.	Repeat.		Repeat.	Repeat.	Repeat.	Repeat.		Reper	Repeat.
				+					_		<u> </u>
Is $a - b\alpha_1$	$ s a - b\alpha_2$	ls a – bo ₃	Is $a - b\alpha_4$	L	Is $a - b\alpha_1$	ls a – bo ₂	Is $a - b\alpha_1$	Is $a - b\alpha_4$		$\ln a - b\alpha_1$	15 4 200
smooth?	smooth?	smooth?	smooth?		smooth?	smooth?	smooth?	smooth?		smooth?	smooth
If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.	1	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) right.	If so, store. Send (a, b) down.		If so, store. Send (a, b) right.	If so, store. Send (a, b) right
Repeat. Is a - bos	Repeat. Is a - bos	Repeat. Is a - boy	Repeat. Is a - bog		Repeat. Is a - bos	Repeat. Is a - bos	Repeat. Is a - boy	Repeat.		Repeat. Is a - bos	Repeat. Is a - boxs
smooth? If so, store,	smooth?	smooth?	smooth?		smooth? If so, store,	smooth? If so, store,	smooth?	smooth?		smooth?	smooth? If so, store,
Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a h) un	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.
Repeat. Is a - boy	Repeat. Is a - bass	Repeat. Is a - bo ₁₁	Repeat. Is a - bo ₁₂	11	Repeat. Is a - bay	Repeat. Is a - ba ₁₀	Repeat. Is a - bo ₁₁	Repeat. Is a - boss	11	Repeat. Is a - bay	Repeat. Is a - ba ₁₀
smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,
Send (a, b) right.	Send (a, b) right.	Send (a, b) right.	Send (a, b) down.		Send (a, b) right.	Send (a, b) right.	Send (a, b) right.	Send (a, b) down.		Send (a, b) right.	Send (a, b) right
Repeat. Is a - bo ₁₃	Repeat. Is a - bass	Repeat. Is a - bo ₁₅	Repeat. Is a - bo ₁₆		Repeat. Is a - bass	Repeat. Is a - ba ₁₄	Repeat. Is a - bo ₁₅	Repeat. Is a - bo ₁₆		Repeat. Is $a - b\alpha_{13}$	Repeat. Is a - ba ₁₄
smooth? If so, store,	smooth? If so, store,	smooth? If so, store,	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,	smooth? If so, store.	smooth? If so, store,		smooth? If so, store,	smooth? If so, store,
Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up.	Send (a, b) left.
Repeat.	Repeat.	Repeat.	Repeat.	1	Repeat.	Repeat.	Repeat.	Repeat.	1	Repeat.	Repeat.
				1					1		
Is a - bos smooth?	Is a - bo ₂ smooth?	Is a - bog smooth?	Is a - box smooth?	1	ls a – ba ₁ smooth?	ls a – bo ₂ smooth?	Is a - boy smooth?	ls a - bra smooth?	11	ls a – bα ₁ smooth?	Is a - bog smooth?
If so, store.	If so, store.	If so, store	If so, store.		If so, store	If so store	If so, store,	If so, store.		If so, store.	If so, store
Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	11	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.	11	Send (a, b) right. Repeat.	Send (a, b) right Repeat.
Is $a = b\alpha_5$	Is $a - b\alpha_6$	Is a - boy	Is $a - b\alpha_k$	11	Is $a - b\alpha_5$	$\ln a - b\alpha_6$	Is $a = b\alpha_7$	ls a – bog	11	$\ln a - b\alpha_5$	Is $a - b\alpha_6$
smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store.
Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.
Is a - boy smooth?	Is a - boso smooth?	ls a - bo ₁₁ smooth?	ls a - boss smooth?	í l	ls a - bay smooth?	ls a - bass smooth?	Is a - boin smooth?	Repeat. Is a - box smooth?	1 í	ls a - bog smooth?	Is a - boso smooth?
If so, store.	If so, store.	If so, store.	If so, store.		If so, store.	If so, store.	If so, store.	If so, store.		If so, store.	If so, store.
Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.		Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) right. Repeat.	Send (a, b) down. Repeat.		Send (a, b) right. Repeat.	Send (a, b) right Repeat.
Is $a = b\alpha_{13}$	Is a - ba ₁₄	Is $a - b\alpha_{15}$	Is $a = b\alpha_{16}$	11		Is $a - b\alpha_{14}$	$b a - b \alpha_{15}$		1 !	$\ln a - b\alpha_{13}$	$b_1 a = b\alpha_{14}$
smooth?	smooth?	smooth? If so, store,	smooth? If so, store,		smooth? If so, store.	smooth? If so, store,	smooth? If so, store.	smooth? If so, store.		smooth? If so, store.	smooth? If so, store,
If so, store.	If so, store.										
Send (a, b) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.		Send (a, b) up. Repeat	Send (a, b) left. Reneat	Send (a, b) left. Reneat	Send (a, b) left. Reneat		Send (a, b) up. Reneat	Send (a, b) left. Reneat
If so, store. Send (a, b) up. Repeat.	If so, store. Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.	Send (a, b) left. Repeat.		Send (a, b) up. Repeat.	Send (a, b) lef Repeat.



	1		
Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$
smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) down.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) down
Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat.
Is a - box Is a - box Is a - box Is a - box smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	Is a - box Is a - box Is a - box Is a - box smooth? smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	ls a − box ls a − box ls a − box smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	Is a - box Is a - box Is a - box Is a - box smooth? smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.
Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.
Is $a - b\alpha_8$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth? smooth?	In the part is a - $b\alpha_{10}$ is a - $b\alpha_{10}$ is a - $b\alpha_{11}$ is a - $b\alpha_{12}$ smooth? Smooth? Smooth?	repeat. repeat. repeat. repeat. Is $a - b\alpha_0$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth?	repeat. Repea
If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Stend (a, b) right. Send (a, b) right. Send (a, b) right.	Smooth? Smooth? Smooth? Smooth? If so, store. If so, store. If so, store. If so, store Send (a, b) right. Send (a, b) right. Send (a, b) down.	Send (a, b) right. Send (a, b) right. Send (a, b) right.
Sand (a, b) right. Sand (a, b) right. Sand (a, b) right. Sand (a, b) down. Repeat. Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_{12}$ Is $a - b\alpha_{13}$ Is $a - b\alpha_{14}$	Sand (a, b) right. Repeat. Repeat. Repeat. Repeat. Is $a - brow$. Is $a - brow$.	Send (a, b) right. Send (a, b) right. Send (a, b) cover. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.	Send (a, b) ngm. Send (a, b) ngm. Send (a, b) ngm. Send (a, b) ngm. Send (a, b) down Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_{11}$ Is $a - b\alpha_{12}$ Is $a - b\alpha_{12}$.
$Is a - Do_{13}$ $Is a - Do_{16}$ $Is a - Do_{16}$ smooth? $smooth$? $smooth$? $smooth$? $smooth$?	is a - Borg is - Borg is - Borg is a - Borg is a - Borg is a - Borg is a - Bor	is a - Ditat is a	smooth? smooth? smooth? smooth? If a show if a
Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.
$ \mathbf{s}_{i} \mathbf{a} - \mathbf{b} \alpha_{i} $		$ \mathbf{s}_{i}\mathbf{a} - b\alpha_{i} $	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_6$
smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store,	smooth? smooth? smooth? smooth? If so, store, If so, store	smooth? smooth? smooth? smooth? If so, store. If so, store.	smooth? smooth? smooth? smooth? smooth?
Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) adven. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) r	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) right. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) down Repeat. Repeat. Repeat.
Is a - box Is a - box Is a - box smooth?	$\frac{1}{10000000000000000000000000000000000$	Is $a - b\alpha s$ is $a - b\alpha s$ is $a - b\alpha s$ is $a - b\alpha r$ is $a - b\alpha s$	Is $a - box$
If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.	Smooth Smooth Smooth Smooth Smooth I So, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left.	If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.
Repeat. Repeat. Repeat. Repeat. Repeat. $ \mathbf{k}_{i} - \mathbf{k}_{i\gamma} \rangle$	Repeat. Repeat. Repeat. Repeat. Repeat. Repeat. $ s a - b_{011}$	Repeat. Repeat. Repeat. Repeat. Repeat. Repeat. $ \mathbf{k}_{i} - b_{0:i} $	Repeat. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.
smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store,	smooth? smooth? smooth? smooth? smooth? If so, store, If s	smooth? smooth? smooth? smooth? If so, store. If so, store.	smooth? smooth? smooth? smooth? smooth? lf so store. If so store. If so store. If so store.
Send (a, b) right. Send (a, b) r	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) right. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) down Repeat. Repeat. Repeat.
respect. respect. respect. respect. respect. respect. Is $a - b\alpha_{13}$ Is $a - b\alpha_{24}$ Is $a - b\alpha_{15}$ Is $a - b\alpha_{16}$ smooth? smooth? smooth?	Is $a - b\alpha_{11}$ is $a - b\alpha_{14}$ is $a - b\alpha_{16}$ is $a - b\alpha_{16}$ smooth? smooth?	repeat. repeat. repeat. Is $a - b\alpha_{11}$ Is $a - b\alpha_{16}$ Is $a - b\alpha_{16}$ smooth? smooth? smooth?	repeat. repeat. repeat. repeat. Is $a - b\alpha_{11}$ is $a - b\alpha_{16}$ is $a - b\alpha_{16}$ is $a - b\alpha_{16}$ smooth? smooth? smooth?
If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	Sted (a, b) up. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.
Repeat. Repeat. Repeat. Repeat. Repeat.	peat. Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat.
$ s _a - b\alpha_1$ $ s _a - b\alpha_2$ $ s _a - b\alpha_3$ $ s _a - b\alpha_4$ $ s _a - b\alpha_4$ $ s _a - b\alpha_4$ $ s _a - b\alpha_4$ $ s _a - b\alpha_4$	$ \mathbf{k} _{a} - b\alpha_1$ $ \mathbf{m}_{a} - b\alpha_2$ $ \mathbf{k} _{a} - b\alpha_4$ $ \mathbf{m}_{a} - b\alpha_4$ $ \mathbf{m}_{a} - b\alpha_4$ $ \mathbf{m}_{a} - b\alpha_4$ $ \mathbf{m}_{a} - b\alpha_4$	$ s a - b\alpha_1$ $ s a - b\alpha_2$ $ s a - b\alpha_3$ $ s a - b\alpha_4$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$ smooth? smooth? smooth? smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) root.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down
Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.	Repeat. Repea	Repeat. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.
smooth? smooth? smooth? smooth? If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? smooth?
Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) left. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
Is $a - b\alpha_0$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_1$ is $a - b\alpha_2$ is $a - b\alpha_3$ is a	Is $a - b\alpha_0$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ is $a - b\alpha_{21}$ is $a - b\alpha_{22}$ is $a - b\alpha_{21}$ is $a - b\alpha_{22}$	Is $a - b\alpha_0$ is $a - b\alpha_{20}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth? smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If So, Store. If	If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right.
Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat. Repeat. Is a - box.	Repeat. Repeat. Repeat. Repeat. is a - box is a - box is a - box is a - box
smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store,	smooth? smooth? smooth? smooth? If so, store, If so, store	smooth? smooth? smooth? smooth? smooth?	smooth? smooth? smooth? smooth? smooth? lf so.store. If so.store. If so.store. If so.store.
Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Senters, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.
$ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_1$ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_2$ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_2$ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_2$	$ s a - b\alpha_1 $ $ s a - b\alpha_2 $ $ s a - b\alpha_3 $ $ s a - b\alpha_4 $	$ \mathbf{i} \cdot \mathbf{a} - b\alpha_1 $ $ \mathbf{i} \cdot \mathbf{a} - b\alpha_2 $ $ \mathbf{i} \cdot \mathbf{a} - b\alpha_3 $ $ \mathbf{i} \cdot \mathbf{a} - b\alpha_4 $	Is a - box Is a - box Is a - box
smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store, If so, store,	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store, If so, store	smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store, If so, store,
Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down Repeat. Repeat. Repeat.
Is $a - b\alpha_5$ Is $a - b\alpha_6$ Is $a - b\alpha_7$ Is $a - b\alpha_8$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_5$ is $a - b\alpha_6$ is $a - b\alpha_7$ is $a - b\alpha_8$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_5$ is $a - b\alpha_6$ is $a - \alpha_7$ is $a - b\alpha_8$ smooth? smooth? smooth?	Is a - bos Is a - bos Is a - bos Is a - bos smooth? Smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.
Repeat. Repeat. Repeat. Repeat. Is a - bα ₀ Is a - bα ₁₀ Is a - bα ₁₁ Is a - bα ₁₂	Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_0$ Is $a - b\alpha_{10}$ Is $a - b\alpha_{11}$ Is $a - b\alpha_{12}$	Repeat. Repeat. Repeat. Reset. Is $a - b\alpha_0$ Is $a - b\alpha_{10}$ Is $a - b\alpha_{11}$ Is $a - b\alpha_{12}$	Repeat. Repeat. Repeat. Repeat. Is a - bα ₁₀ Is a - bα ₁₀ Is a - bα ₁₁ Is a - bα ₁₂
smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store. If so, store.
Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down. Repeat. Repeat. Repeat. Repeat.	Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down Repeat. Repeat. Repeat. Repeat.
Is $a - b\alpha_{13}$ Is $a - b\alpha_{14}$ Is $a - b\alpha_{15}$ Is $a - b\alpha_{16}$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_{13}$ is $a - b\alpha_{14}$ is $a - b\alpha_{15}$ is $a - b\alpha_{26}$ smooth? smooth? smooth? smooth?	$\frac{ s a - b\alpha_{13}}{smooth?} \frac{ s a - b\alpha_{16}}{smooth?} \frac{ s a - b\alpha_{26}}{smooth?}$	is $a - b\alpha_{13}$ is $a - b\alpha_{26}$ is $a - b\alpha_{26}$ is $a - b\alpha_{36}$ smooth? is smooth? is smooth? smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	If so, store. If so, store. If so, store. If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	If so, store. Send (a, b) up. Send (a, b) left. Send (a, b) left.	If so, tree. If so, store. If so, store. If so, store. If so, store. Send (a, b) Send (a, b) left. Send (a, b) left. Send (a, b) left.
Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat.	Repeat. Repeat. Repeat. Repeat.
Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_4$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_1$ smooth? Is $a - b\alpha_2$ smooth? Is $a - b\alpha_3$ smooth? Is $a - b\alpha_4$ smooth? Is $a - b\alpha_4$ smooth?	Is $a - b\alpha_1$ smooth? Is $a - b\alpha_2$ smooth? Is $a - b\alpha_3$ smooth? Is $a - b\alpha_4$ smooth? Is $a - b\alpha_4$	Is $a - b\alpha_1$ Is $a - b\alpha_2$ Is $a - b\alpha_2$ Is $a - b\alpha_3$ Is $a - b\alpha_6$ smooth? smooth? smooth? smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) down.	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) down
Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_5$ Is $a - b\alpha_6$ Is $a - b\alpha_7$ Is $a - b\alpha_8$	Repeat. Repeat. Repeat. Repeat. $ s a - b\alpha_5$ $ s a - b\alpha_6$ $ s a - b\alpha_7$ $ s a - b\alpha_8$	Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_5$ Is $a - b\alpha_6$ Is $a - b\alpha_7$ Is $a - b\alpha_8$	Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_6$ Is $a - b\alpha_8$ Is $a - b\alpha_8$ Is $a - b\alpha_8$
smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.	smooth? smooth? smooth? smooth? If so, store. If so, store. If so, store.
Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.
Is $a - b\alpha_0$ Is $a - b\alpha_{10}$ Is $a - b\alpha_{11}$ Is $a - b\alpha_{12}$ smooth? smooth? smooth? smooth?	$ s a - b\alpha_0$ $ s a - b\alpha_{10}$ $ s a - b\alpha_{11}$ $ s a - b\alpha_{12}$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_9$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth? smooth? smooth?	Is $a - b\alpha_9$ is $a - b\alpha_{10}$ is $a - b\alpha_{11}$ is $a - b\alpha_{12}$ smooth? smooth? smooth? smooth?
If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down.	If so, store. If so, store. If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) down
Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_{13}$ Is $a - b\alpha_{16}$ Is $a - b\alpha_{16}$ Is $a - b\alpha_{16}$	Repeat. Repeat. Repeat. Repeat. $ s a - b\alpha_{13} $ $ s a - b\alpha_{24} $ $ s a - b\alpha_{25} $ $ s a - b\alpha_{26} $	Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_{10}$ Is $a - b\alpha_{16}$ Is $a - b\alpha_{16}$ Is $a - b\alpha_{16}$	Repeat. Repeat. Repeat. Repeat. Is $a - b\alpha_{13}$ Is $a - b\alpha_{26}$ Is $a - b\alpha_{15}$ Is $a - b\alpha_{26}$
smooth? smooth? smooth? smooth?	smooth? smooth? smooth? smooth? If so, store, If so, store, If so, store,	smooth? smooth? smooth? smooth?	smooth? smooth? smooth? smooth?
Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat.	Send (a, b) up. Send (a, b) left. Send (a, b) left. Send (a, b) left. Repeat. Repeat. Repeat. Repeat.
Repeat. Repeat. Repeat. Repeat.			

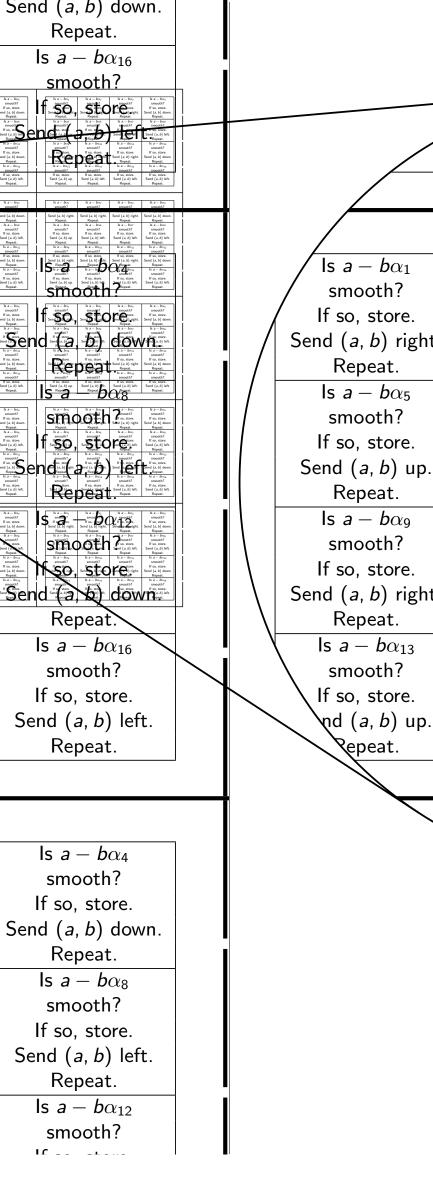
	attioutin.	
	If so, store.	
snt.	Send (<i>a</i> , <i>b</i>) right.	Ser
<i>.</i>	Repeat.	
$b \alpha_5$	Is $a - b \alpha_6$	ls a
oth?	smooth?	smod
, store.	If so, store.	If so, s
(<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (a, k
epeat.	Repeat.	Repea
$-b\alpha_9$	Is $a - b \alpha_{10}$	ls <i>a</i> – /
oth?	smooth?	smod
store.	If so, store.	If so,
∖ right.	Send (<i>a</i> , <i>b</i>) right.	Send (
	Repeat.	
	Is $a - b \alpha_{14}$	
	smooth?	

$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:heat} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$
$\label{eq:results} \begin{array}{ c c c c c c } \hline h = -b v_1 & h = $	$\label{eq:second} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c } \hline h = -h v_1 & h = -h $
$\label{eq:results} \begin{array}{ c c c c c c } \hline h = -h v_1 & h = $	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c } \hline Is a - bay, & Is a$	$\label{eq:results} \begin{array}{ c c c c c c } \hline h = -h y_1 & h = $
$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c } \hline h = -hr_1 & h = -hr_2 & h = -hr_2 & ancohr \\ \hline monohr & monohr & ancohr & ancohr \\ \hline monohr & monohr & ancohr & ancohr \\ \hline monohr & monohr & ancohr & ancohr \\ \hline monohr & h = -hr_1 & h = -hr_2 & $	$\label{eq:result} \begin{array}{ c c c c c } \hline h = -hr_1 & h = -hr_2 & h = -$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:results} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$

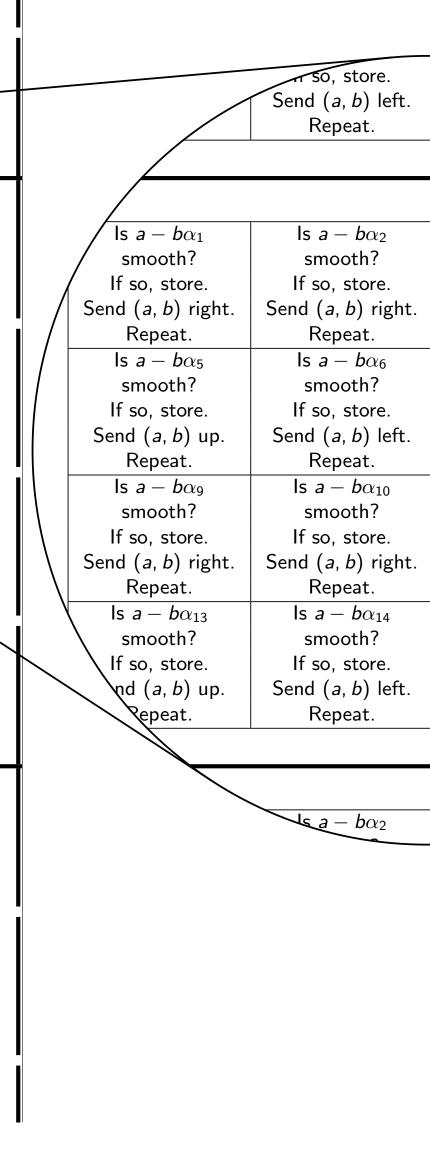


	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.			
	Repeat.	Repeat.	Repeat.			
	Is $a - b \alpha_{13}$	Is $a - b lpha_{14}$	Is $a - b lpha_{15}$			
	smooth?	smooth?	smooth?			
	$ \begin{array}{c c} \mathbf{k} \mathbf{z} - \mathbf{k} \mathbf{v}_1 \\ \mathbf{s} = \mathbf{k} \mathbf{v}_1 \\ \mathbf{s} \mathbf{v}_1 \\ \mathbf{s} \mathbf{v}_1 \\ \mathbf{s} = \mathbf{k} \mathbf{v}_1 \\ \mathbf{s} \mathbf{v}_1 \\ \mathbf$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c c} is a-br_1 \\ monoth? \\ if o_{0}, torow \\ th \\ strength \\ Strengt \\ Strengt \\ Strengt \\ Strength \\ Strength \\ Strength \\ Str$	Is a sm If so ht. Send (a Re		
	$ \begin{array}{c} \mathbf{k} a - \mathbf{k} \mathbf{v} \\ \mathbf{k} a - \mathbf{k} \mathbf{v} \\ \mathbf{k} \mathbf{v} = \mathbf{k} \mathbf{v} \\ \mathbf{k} \mathbf{k} \\ \mathbf{k} \mathbf{k} \\ \mathbf{k} \mathbf{k} \\ \mathbf{k} \\ \mathbf{k} \mathbf{k} \\ $	$\begin{array}{cccc} a_{1} & & & & a - b x \\ b_{2} & & & & b_{3} - b x \\ c_{1} & & & & b_{3} - b x \\ c_{2} & & & & b_{3} - b x \\ c_{2} & & & & b_{3} - b x \\ c_{3} & & & & b_{3} - b $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	Is a sm If so t. Send (Re Is a		
	$\label{eq:smooth} \begin{array}{c} \mbox{smooth}^7 & \mbox$	$\label{eq:constraint} \begin{array}{c} monoth? \\ monoth?$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	sm If so ht. Send (<i>a</i> Re Is <i>a</i>		
	If yo, tore. If yo, tore.<	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$. If so, store,	t. Send (Re		
	Is a - bro; smooth? Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b)		Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_1$ Is $a - bx_2$ Is $a - bx_2$ Is $a - bx_1$ Is $a - bx_2$ Is $a -$	Is a sm ht. Send (a		
	Repart. Repart. <t< td=""><td>box is $a = box$ is $a = box$ is $b = box$ h7 smooth? if box, tore. If bo</td><td>$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$</td><td>t. Send (Re If so Re Is a sm</td></t<>	box is $a = box$ is $a = box$ is $b = box$ h7 smooth? if box , tore. If bo	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	t. Send (Re If so Re Is a sm		
	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	t Repeat Repeat Repeat Repeat Repeat Repeat Repeat Repeat $a = b\alpha_{14}$ $b = b\alpha_{12}$ $b = a = b\alpha_{16}$	H Starting West, stores, store West, stores, store West, store, store West, store West, store<	ht. Send (J Re Is a sm		
	Man atom Sand (A) by Sand (A) int Repeat. Served (A) int Served (A	n, Brantzen, Bra	t. See stores to store a store of the store	t. Send (
	$ \begin{array}{c c} k = -kr_{1} \\ \hline k = -kr_{2} \\ \hline m codth^{2} \\ m codth \\ m codt} \\ \hline k = -kr_{2} \\ \hline m codth^{2} \\ m codth^{2} \\ m codth \\ m co$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c} & i_{2} - b_{2} \\ \hline \\ & modh? \\ M \\ \\ $	ls a sm If so ht. Send (a Re		
	$\label{eq:second} \begin{array}{ c c c c } \hline & & & & & & & & & & & & & & & & & & $	$\begin{array}{cccc} \mathbf{x}_{1} & \mathbf{x}_{2} - \mathbf{x}_{1} \\ \mathbf{x}_{2} & \mathbf{x}_{3} \\ \mathbf{x}_{1} & \mathbf{y}_{2} \\ \mathbf{x}_{2} \\ \mathbf{y}_{2} \\ \mathbf{y}_{3} \\ \mathbf{y}_{4} \\ \mathbf{y}_{5} \\ \mathbf{y}_{4} \\ \mathbf{y}_{5} \\ \mathbf{y}_{4} \\ \mathbf{y}_{5} \\ \mathbf$	$\begin{array}{c} \begin{array}{c} \mathbf{i} \ a - b \mathbf{v}_{1} \\ \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \\ \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \\ \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \\ \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \\ \mathbf{s} \ a - b \mathbf{s} \ a - b \mathbf{s} \\ \mathbf{s} \ a - b \mathbf{s} \ a$	t. Send (
	smooth? smooth? <t< td=""><td>$\begin{array}{cccccccccccccccccccccccccccccccccccc$</td><td>smooth? smooth? smooth? smooth? If 50, store, If 50, store, If 50, store, If 50, store, ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right.</td><td>sm If so ht. Send (a Re Is a</td></t<>	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	smooth? smooth? smooth? smooth? If 50, store, If 50, store, If 50, store, If 50, store, ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. ht. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right. Smd (a, b) right.	sm If so ht. Send (a Re Is a		
	Send (a) pp. Send (a) list Sen	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	t See store II to store II to store See (a) II to store II to store See (a) II to store See (a) II to store (b) II to store See (c) II to store (c) II to store See (c	t. Send (
	$\begin{tabular}{ c c c c c c c } \hline & & & & & & & & & & & & & & & & & & $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c} \text{is } a - b c_1 \\ \text{smooth} \\ \text{if } a_n \text{const} \\ \text{const} \\ \text{if } a_n \text{const} \\ \text{const} \\ \text{if }$	ht. Send (J Re		
	$ \begin{array}{c} \mathbf{k} \mathbf{z} - \mathbf{b} \mathbf{y} \\ \mathbf{f} \mathbf{x} - \mathbf{b} \mathbf{x} \\ \mathbf{f} \mathbf{x} , \mathbf{c} \mathbf{x} \\ \mathbf{s} \mathbf{g} \mathbf{g} \mathbf{x} \\ \mathbf{s} \mathbf{g} \mathbf{g} \mathbf{g} \mathbf{x} \\ \mathbf{k} \mathbf{z} - \mathbf{b} \mathbf{x} \\ \mathbf{s} \mathbf{g} \mathbf{g} \mathbf{g} \mathbf{g} \mathbf{g} \mathbf{g} \mathbf{g} g$	$ \begin{array}{c c} \mathbf{x}_{1} & \mathbf{z}_{2} - \mathbf{b}_{1} \\ \mathbf{x}_{2} \\ \mathbf{x}_{3} \\ \mathbf{x}_{4} \\ \mathbf{x}_{5} \\ \mathbf$	It $a - bry$ smooth? If a_0 , store, th. Smooth? Sm	t. Send (Re		
	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c} \begin{array}{c} 1 & a = b + a \\ 1 & a = b + a \\ 1 & c = b + a \\ c = b + b \\ c = b \\ c$	$\begin{array}{c c} & a = a_{12} \\ a = a_{12} \\ a = a_{12} \\ b = b_{12} \\ b = b_{$	ht. Send (a Re Is a		
	smooth? If so, store. Sind (a, b) up. Repeat. Sind (a, b) up. Repeat. Sind (a, b) up. Sind (a, b) up.	The smooth? The Brooth? The Steer (a, b) lift. Repu	t Sind (a, b) left. Sind (a, b) left. Repeat. Repeat. Repeat.	t. Send (
	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c} Is a - bv_1 \\ monoth \\ If a - bv_1 \\ monoth \\ If a - bv_1 \\ If $	Is a sm If so ht. Send (a		
	$ \begin{array}{c c} \mathbf{i}, \mathbf{a} - b\mathbf{x}_1 \\ \text{smooth7} \\ \mathbf{smooth7} $	h_{1} $h_{2} - b_{1}$ $h_{3} - b_{2}$ $h_{3} - b_{2}$ $h_{3} - b_{2}$ $h_{4} - b_{2}$ $h_{5} - b_{2}$ $h_{5} - b_{2}$ $h_{5} - b_{2}$	$ \begin{array}{c c} \mathbf{i} s = -b \mathbf{r}, \\ \mathbf{i} s = -b \mathbf{r}$	t. Send (
	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	T smooth? The smooth? The store for smooth? The store for smooth? The store for store for store for smooth? The store for store for store for smooth? The store for store for store for store for smooth? The store for sto	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Is a sm If so ht. Send (a Re Is a		
	Sterring and the standard stan	$ = \underbrace{Send_{aut}}_{pure transfer trans$	t senseriar right to some	t. Send Re		
	Repeat.	Repeat.	Repeat.			
Ì	Is $a - b\alpha_{13}$	Is $a - b \alpha_{14}$	Is $a - b\alpha_{15}$			
	smooth?	smooth?	smooth?			
	If so, store.	If so, store.	If so, store.			
	Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.			
	Repeat.	Repeat.	Repeat.			
I		1		1		

Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.
Repeat.	Repeat.	Repeat.
Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$
smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
Repeat.	Repeat.	Repeat.
Is $a - b\alpha_9$	Is $a - b lpha_{10}$	Is $a - b\alpha_{11}$
smooth?	smooth?	smooth?
16	IC	



		Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
		Repeat.	Repeat.	Repeat.	Repeat.
		Is $a - b\alpha_{13}$	Is $a - b \alpha_{14}$	Is $a - b \alpha_{15}$	Is $a - b\alpha_{16}$
JUTT		smooth?	smooth?	smooth?	smooth?
		$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	le a ber le a hor le a hor le a hor le a hor	h $k_2 - k_{12}$ $k_3 - k_{12}$ $k_4 - k_{12}$ $k_5 - k_{1$	It a host It a host It a host It a host
store. 🛸		under the second	$ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_0 $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{10} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} $	is $a - b\alpha_{12}$ is $a - b\alpha_{22}$ is $a - b\alpha_{12}$ is $a - b\alpha_{12}$
		$ \begin{array}{c c} I' \sigma_{12}, \sigma_{12}, \sigma_{13}, \sigma$	Baseline Userable Dissolution Dissolution Dissolution Baseline Userable Dissolution Dissolution Dissolution Baseline Algorithm Dissolution Dissolution Dissolution Baseline Algorithm Dissolution Dissolution Dissolution Dissolution Baseline Algorithm Dissolution Dissolution Dissolution Dissolution Dissolution Baseline Algorithm Dissolution	If U so, down If and Res If a stress If a stress If a stress interpret. bing Res bing Res </td <td>$\begin{array}{ c c c c c c c c c c c c c c c c c c c$</td>	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
b) right.	Sèr	Image: strength of the	Repeat Repat. Rep. Rep. Rep.	intermediation intermediation intermediation intermediation 1 ista - broj ista - broj ista - broj ista - broj 1 ista - broj ista - broj ista - broj ista - broj	Name Nam Name Name
eat.			$\label{eq:constraints} \begin{array}{cccc} \mbox{stream} & \mbox{stream}$	ight Send (a, b) right. Send (a, b) right. <td>$\label{eq:second} \begin{array}{c} \mbox{second} & \mbox$</td>	$\label{eq:second} \begin{array}{c} \mbox{second} & \mbox$
eat.			$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{cccc} \operatorname{serf}(a,b) \operatorname{str.} & \operatorname$	t. Serie (a, b) left. Repeat. Repeat. Repeat. Is a - bray. series of the analysis of the a - bray. Is a - bray. series of the a - bray. Is a - bray. Is a - bray. Series of the a - bray. If Sea, the a - bray. Series of the a - bray. If Sea, the a - bray. Series of the a - bray.
hou		$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	AL Send (24) right. Son (24) 30 min. Report. R	Star (L a) Johns, Star (L a) Johns, Star (L a) Johns, Star (L a) Johns, Ha - Drug, second? Bit a - Drug Bit a - Drug Bit a - Drug Bit a - Drug With a - Drug Bit a - Drug With a - Drug Bit a - Drug
$-D\alpha_6$	ls <i>a</i>	Surf (A) yin Surf (A) yin<	Appendix Stard (A) Mr.	Smooth : we set to se	Sord (a, b) th, Sord (a, b
- $m{b} lpha_6$ oth?	smo	A sense i de la d	and Charles and Charles I and	lea hou lea hou lea hou lea hou lea hou	ka hou ka hou ka hou ka hou
	smo ू	Sec. Try Tr	Direction Direction <thdirection< th=""> <thdirection< th=""> <thd< td=""><td>$\begin{array}{c} \begin{array}{c} \begin{array}{c} \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} - \mathbf{h}_{2} \\ \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} \\ \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_$</td><td></td></thd<></thdirection<></thdirection<>	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} - \mathbf{h}_{2} \\ \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} \\ \mathbf{h}_{2} \rightarrow \mathbf{h}_{2} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_{3} \rightarrow \mathbf{h}_{3} \\ \mathbf{h}_$	
store.	If so, s	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	smooth? smooth? smooth? smooth?	Report. Report. <t< td=""><td></td></t<>	
51010.			ha - bit - b	13 a - bry most? 1 a - bry 1 a	
b) left.	Send (a, k	$ \begin{array}{ c c c c c } \hline sord(a,b) a (a,b) & Sord(b,b) & So$	$ \begin{array}{c} \operatorname{ide}(A) \partial_{\mu} h_{1} & \operatorname{Sord}(A) \partial_{\mu} h_{2} & \operatorname{Sord}(A) \partial_{$	Is a - box Is a - box Is a - box	
•			$ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_0 $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{10} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $ $ \mathbf{s} \cdot \mathbf{a} - \mathbf{b} \alpha_{12} $	Repart and Particle P	$ \mathbf{s} \mathbf{a} - \mathbf{b}\alpha_{12} $ $ \mathbf{s} \mathbf{a} - \mathbf{b}\alpha_{0} $ $ \mathbf{s} \mathbf{a} - \mathbf{b}\alpha_{12} $ $ \mathbf{s} \mathbf{a} - \mathbf{b}\alpha_{12} $
eat.	Repea	K = A Bright K = A Bright <thk =="" a="" bright<="" th=""></thk>		1 1 - Кису - К	
	Керес	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		$\label{eq:rescaled_rescale} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	Ib a - Int, second? W a - Sec W a - Sec W a - Sec If M, abox If M a - Sec Int a - Sec Int a - Sec Int a - Sec Sec(A) down If M a - Sec Sec(A) down If M a - Sec If M a - Sec If M a - Sec
$m{b}lpha_{10}$	l Is <i>a</i> – I	Sind (a, b) ript. Sind (a, b) ript. Sind (a, b) ript. Sind (a, down, Repeat. Repeat. Repeat. Repeat. Repeat. $\frac{1}{16} a - bx_1$ is $a - bx_1$ is $a - bx_1$ is $a - bx_1$ is $a - bx_1$ in $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - bx_1$ if $a - bx_1$ if $a - bx_1$ is $a - bx_1$ if $a - $	$ \begin{array}{c} \operatorname{sind}(a,b)\operatorname{spit}(a,b)\operatorname{sind}(a,b)\operatorname{spit}(b,$	Repeat. Repeat. Repeat. Repeat. Repeat.	s Sind (x) shown Repair $b = -bx_1$ $b = -bx_1$ b
		response response response response			ka ha ka ka ka ha ka ha
oth?	smod	$S_{\text{result}}^{\text{h}a-bray} = S_{\text{result}}^{\text{h}a-bray} = S_{\text$	nord i monti i	i a - brig i agosti t Storen Gat t Storen	$\begin{array}{c} \begin{array}{c} \mathbf{h} = -b \mathbf{x}_{1} \\ \mathbf{s} = -b \mathbf{x}_{1}$
		Repeat.	Repeat.	Repeat.	Repeat.
store.	lf so,/	Is $a - b\alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$
		smooth?	smooth?	smooth?	smooth?
b) right.	Send (/	If so, store.	If so, store.	If so, store.	If so, store.
eat.	y	Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.
eat.		Repeat.	Repeat.	Repeat.	Repeat.
ha					
$Dlpha_{14}$					
$blpha_{14}$ oth?					
JUII!		Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$
		smooth?	smooth?	smooth?	smooth?
		If so, store.	If so, store.	If so, store.	If so, store.
		Send (a, b) right.	(, -	Send (a, b) right.	Send (<i>a</i> , <i>b</i>) down.
		Repeat.	Repeat.	Repeat.	Repeat.
		Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$
		smooth?	smooth?	smooth?	smooth?
		If so, store.	If so, store.	If so, store.	If so, store.
		Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.
		Repeat.	Repeat.	Repeat.	Repeat.
		Is $a - b\alpha_9$	Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$	Is $a - b\alpha_{12}$
		smooth?	smooth?	smooth?	smooth?

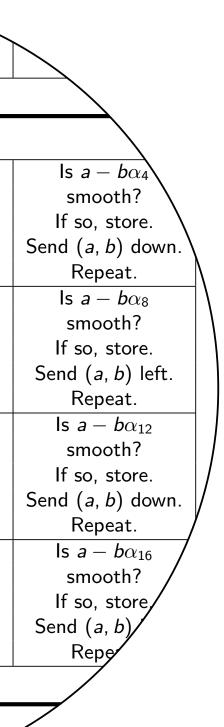


	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.					
	Repeat.	Repeat.	Repeat.	Repeat.	_				
	ls $a - b\alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$		8			
	smooth?	smooth?	smooth?	smooth?					
	I so the provided for the second seco	$ \begin{array}{c} -a_{11}^{k} & h_{2}-b_{22}^{k} \\ h_{2}-b_{22}^{k} & h_{2}-b_{22}^{k} \\ h_{2}-b_{2}^{k} & h_{2}-b_{2}^{k} \\ h_{2}-b_{2}^{k}$	Ista-bay	method 1 <td></td> <td></td> <td></td> <td>Send (<i>a</i>, <i>b</i>) left.</td> <td>If so, stor</td>				Send (<i>a</i> , <i>b</i>) left.	If so, stor
	See DC II they be derived by the der	(a d) work Stand (a b) and	the second seco	Seed (a) bits Control (b)				Repeat.	Send (<i>a</i> , <i>b</i>) left. Repeat.
Ser	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	smooth? smooth? smooth? smooth? smooth?					
	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$a - b\alpha_1$ is $a - b\alpha_2$ is $a - b\alpha_1$ is $a - b\alpha_2$ smooth? smooth? smooth? smooth? smooth? smooth?	is a - bo1 is a -	$\begin{tabular}{ c c c c c } \hline & & & & & & & & & & & & & & & & & & $		Ļ			
	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
Is a	Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Non-on-i Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic Sore (a, b) optic	A. Bright Sout (A) right Sout (A) right Sout (A) (ht. Sard (a) right. Sard (a) down Repart. Barrow Barrow Barrow account of the same and the sa	a Sorie (A) about <u>Branch</u> 1 Sorie (A) <u>Branch</u> 1 Sorie (A) <u>Sorie (A) <u>Branch</u> 1 Sorie (A) <u>Sorie (A) <u>Branch</u> 1 Sorie (A) <u>Sorie (A) <u>Branch</u> 1 Sorie (A) <u>Sorie (A) <u>Sorie (A) <u>Branch</u> 1 Sorie (A) <u>Sorie (A) Sorie (A) Sorie (A) Sorie (A) Sorie (A) <u>Sorie (A) Sorie </u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u>			$\int smooth?$	Is $a - b\alpha_2$ smooth?	Is $a - b\alpha_3$ smooth?
	Is a - bry smooth? Is a -	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c} \mathbf{i}_{k,d} - \mathbf{b}_{2k} \\ \mathbf{monoth}^{T} \\ \mathbf{monoth}^{$				If so, store.	If so, store.	If so, store.
smo	$\begin{bmatrix} \frac{h_1 - h_1}{2} & \frac{h_2 - h_2}{2} \\ \frac{h_1 - h_2}{2} & \frac{h_1 - h_2}{2} \\ \frac{h_2 - h_1}{2} \\ \frac{h_1 - h_2}{2} \\ \frac{h_2 - h_1}{2} \\ \frac{h_1 - h_2}{2} \\ \frac{h_2 - h_2}{2} \\ \frac{h_1 - h_2}{2} \\ h_1 - $	$\begin{array}{c} \mathbf{h}_{1} = \mathbf{h}_{2} \\ \mathbf{h}_{2} = \mathbf{h}_{1} \\ \mathbf{h}_{2} = \mathbf{h}_{2} \\ \mathbf{h}_{2} \\ \mathbf{h}_{2} = \mathbf{h}_{2} \\ \mathbf{h}$	$\begin{array}{c c} \mathbf{h}_{a-bry} & \mathbf{h}_{a-bry} \\ \mathbf{h}_{a-bry} & \mathbf{h}_{a-bry}$	$\begin{array}{c} \mathbf{h}_{a} \rightarrow \mathbf{h}_{a} \\ \mathbf{h}_{a} \rightarrow \mathbf{h}_{a} \\ \mathbf{s} \\ $		I,	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.
	$\label{eq:rescaled_rescale} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	h. abbe μ (m. tax. H (m. tax	H B or, core, B B or, co	If is, store, in the			Repeat.	Repeat.	Repeat.
lf so, s	Seef (A) ip See (A) ip	(24) w . Sourd (24) int	h Serie (A) Series Series (A) Series Byond Byo	Sord (a, b) int. Sord (a, b) int.<	1		Is $a - b\alpha_5$ smooth?	Is $a - b\alpha_6$ smooth?	Is $a - b\alpha_7$ smooth?
Send (a, k	School (2010) School ($\begin{array}{c} \operatorname{Supp}_{A} = \operatorname{Supp}_{A} \left(\begin{array}{c} \operatorname{Supp}_{A} \\ S$	$\begin{array}{c} & \mbox{ If } C_{1} \cap \mathcal{G}_{1} \cap \mathcal{G}_{2} \cap \mathcal{G}$	$\label{eq:second} \left[\begin{array}{c} so & f(x) = f(x) \\ respect \\$			If so, store.	If so, store.	If so, store.
		to a manufacture in the second	L and from the state of the sta	3 Second 3 Second <td< td=""><td></td><td></td><td>Send (<i>a</i>, <i>b</i>) up.</td><td>Send (a, b) left.</td><td>Send (a, b) left.</td></td<>			Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (a, b) left.
Repea	H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Stard (A) H a - Any H a - Any H a - Any H a - Any Repart <	norden statung har han har han har	Bit Abit	Norm Series Representation Representa			Repeat.	Repeat.	Repeat.
· · · · · · · · · · · · · · · · · · ·	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\label{eq:rescaled_static} \left \begin{array}{c} \mathrm{i} t a - b v_1 \\ \mathrm{secoth}^2 \\ \mathrm{i} \ \mathrm{secoth}^2 \\ \mathrm{i} \ \mathrm{secoth}^2 \\ \mathrm{i} \ \mathrm{secoth}^2 \\ secot$			Is $a - b\alpha_9$	Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$
ls <i>a</i> – <i>l</i>	Nipsi. reput. rept. rept. rept. <td>ropation repairs and the second seco</td> <td>Instruct. registr. registr.</td> <td>$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$</td> <td></td> <td></td> <td>smooth?</td> <td>smooth?</td> <td>smooth?</td>	ropation repairs and the second seco	Instruct. registr.	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$			smooth?	smooth?	smooth?
	1 1	1 model = mod	immedia immedia <t< td=""><td>Image: State of the s</td><td></td><td></td><td>If so, store.</td><td>If so, store.</td><td>If so, store.</td></t<>	Image: State of the s			If so, store.	If so, store.	If so, store.
smog		a str. (ch pp spect Sendout (a) (a de (ab)) (rigent. agent spect (a) (b) (b) (b) (b) (b) (b) (b) (b) (b) (b					Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) right.
	Repeat.	Repeat.	Repeat.	Repeat.	-	`	Repeat.	Repeat.	Repeat.
If so,/	Is $a - b\alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	$ $ Is $a - b\alpha_{16}$			$\int \mathbf{ls} \mathbf{a} - \mathbf{b} \alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$
Send (/	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.	smooth? If so, store.			smooth?	smooth? If so, store.	smooth? If so, store.
	Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.	Send (a, b) left.			nd (a, b) up.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.
y	Repeat.	Repeat.	Repeat.	Repeat.			Repeat.	Repeat.	Repeat.
							epcut.		
						l			
	Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$]			$b\alpha_2 - b\alpha_2$	Is a – ba
	smooth?	smooth?	smooth?	smooth?					
	If so, store.	If so, store.	If so, store.	If so, store.					
	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.					
	Repeat.	Repeat.	Repeat.	Repeat.					
	Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$					
	smooth?	smooth?	smooth?	smooth?					
	If so, store.	If so, store.	If so, store.	If so, store.					
	Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (a, b) left.	Send (a, b) left.					
	Repeat.	Repeat.	Repeat.	Repeat.	-				
	$ s a - b\alpha_9 $	Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$	ls $a - b\alpha_{12}$					
	smooth?	smooth?	smooth?	smooth?					

Send	d (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.						
	Repeat.	Repeat.	Repeat.	Repeat.						
l	s a – b $lpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b \alpha_{15}$	Is $a - b\alpha_{16}$						
	smooth?	smooth?	smooth?	smooth?	1					
Is a - bos smooth? If so, store. Send (a, b) right Repeat. Is a - bos	s a boy s a boy f if a boy m d S of the second red speak. s a boy is a -	h a - bin,	$ \begin{array}{c} 1 & 1s - br_1 \\ smooth^7 \\ sh \\ s$	$\begin{array}{c} \mathbf{h} \mathbf{a} - \mathbf{b} \mathbf{a} \\ \mathbf{s} \\$				so, store.	If so, stor	
If Send Set	$\begin{array}{c} \begin{array}{c} & \text{is an orbit} \\ \text{if Colore,} \\ \text{fl Colore,} \\ \ fl Colore,} \\ \text{fl Colore,} \\ \ fl Colore,} \\ \text{fl Colore,} \\ \ fl Colore,} \\$	$\begin{array}{c c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \\ \end{array} \end{array} \\ \begin{array}{c} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} $	$ \begin{array}{c} & \text{in cost} \\ & \text{in cost} $	account account <t< th=""><th></th><th></th><th></th><th>Send (<i>a</i>, <i>b</i>) left.</th><th>Send (<i>a</i>, <i>b</i>) left.</th><th></th></t<>				Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	
smooth?	$ \begin{array}{c c} smooth' & smooth' & smooth' \\ \hline If b_k D_{a_k} & If b_k store, & If so, store, \\ \hline Send Price & Price & Price & Price \\ \hline b_k a - bo_{1k} & Free bar, \\ \hline b_k a - bo_{1k} & Free bar, \\ \hline smooth? & smooth? \\ \end{array} $	$ \begin{array}{c} \mbox{incode} in$	$\begin{array}{c} \operatorname{smooth}'_{i} & \operatorname{smooth}'_{i} &$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $				Repeat.	Repeat.	
If so, store. Send (a, b) up. Repeat.	If so, store. If so, store. Send (a, b) bit. Send (a, b) bit. Send (a, b) bit. Repeat. Repeat. Repeat.	If So, Store. See See See See See See See See See S	a. If so, store. If so, store. Mr. S. end (a, b) Wr. S. end (b, b)	If ico, acros. Sond (a,b) olif. Sond (a,b) oli	_					
Is a - bas smooth? Send (a, b) right. Repeat.	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$\label{eq:rescaled} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	Is a - bx1 senseth? L Send (a, b) down. Repeat. Send (a, b) right. Repeat.						
Is a - bos smooth? If so, store. Send (a, b) up. Repeat. Is a - bos		$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	s is a - bor is a - bos is - bos	is a - box is a - box is a - box is a - box smooth7 smooth7 smooth7 smooth7 smooth7 16 or, store. Smrd (a, b) tht. Repeat. Repeat. Repeat. Repeat. Repeat. Repeat.						\backslash
semoch? If so, store. Send (a, b) right. Repeat. Is a - brain semoch?	smooth? smooth? smooth? If so, store. If so, store. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (a, b) right. Send (b) r	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	smooth? smooth? smooth? smooth? smooth? If wo, store.	smooth? smooth? <t< th=""><th></th><th></th><th>Is $a - b\alpha_1$</th><th>Is $a - b\alpha_2$</th><th>Is $a - b\alpha_3$</th><th>Is $a - b\alpha_4$</th></t<>			Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$
If so, store. Send (<i>a</i> , <i>b</i>) up. Repeat.	Smooth Stores St	V So, tabe. (c) & Hup, Send (a) birt. Repeat. Since (b) Hup, Send (a) birt. Repeat. Since (b) Hup, Send (b) Kirt. Repeat. Since (b) Hup, Send (b) Kirt. Repeat. Hiso, store. Since (b) Kirt. Repeat. Hiso, store. Since (b) Kirt. Repeat. Since (b) Kirt. Since (b	t Sind (a, b) lift. Sind (a, b	If ico, store. If ico, store. If ico, store. If ico, store. Sond (a,b) lift. Sond (a,b) post. Sond (a,b) post. If ico, store. If ico, store. Repeat. Simod (a,b) post. Sond (a,b) post. <td< th=""><th>_</th><th></th><th>/ smooth?</th><th>smooth?</th><th>smooth?</th><th>smooth?</th></td<>	_		/ smooth?	smooth?	smooth?	smooth?
Is a - bo ₁ smooth? If so, store. Send (a, b) right Repeat.	$\label{eq:second} \left[\begin{array}{c} \mathbf{k} \ a - b \mathbf{v}_3 \\ \mathbf{s} \ moth \mathbf{k} \\ \mathbf{s} \\ s$	h a - bit, immotify - bit, immotify b a - bit, immotify Is a - bit, immotify Vis open, immit (a b) right. Send (a b) right. Find (a b) right. Find (a b) right. Find (a b) right. Repart. Repart. Repart. Repart. Repart. Repart.	$ \begin{array}{c} \textbf{is} a - b \textbf{v}_1 \\ \textbf{south} \\ so$	Is a - bn; secoth? Is a -			/ If so, store.	If so, store.	If so, store.	If so, store. \setminus
			$ \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \end{array}\\ \end{array}\\ \end{array}\\ \end{array}\\ \end{array}\\ \end{array}\\ \\ \end{array}\\ \\ \end{array}\\ \\ \end{array}\\ \\ \end{array}\\$	$ \sum_{n=1}^{n} \sum_{n=1}^{n} nd \sum_{n=1}^{n} \sum_{n=1}^{n} \frac{1}{n} \sum_{n$		/	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) down.
smooth? If so, store. Send (a, b) right. Repeat. Is a - bv _{1k} smooth?	simooth/ if mooth/ if simooth/ if simooth/ <t< th=""><th>$\begin{array}{c} \operatorname{smooth}^{\prime} & smoot$</th><th>$\begin{array}{c c c c c c c c c c c c c c c c c c c$</th><th>smooth? smooth? <t< th=""><th></th><th></th><th>Repeat.</th><th>Repeat.</th><th>Repeat.</th><th>Repeat.</th></t<></th></t<>	$\begin{array}{c} \operatorname{smooth}^{\prime} & smoot$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	smooth? smooth? <t< th=""><th></th><th></th><th>Repeat.</th><th>Repeat.</th><th>Repeat.</th><th>Repeat.</th></t<>			Repeat.	Repeat.	Repeat.	Repeat.
If so, store. Send (a, b) up. Repeat.	If so, store. Send (a, b) left. Send (b, b) left	If So, tables. If Do, Store. If Do, Store. If Do, Store. Soc. Aster. Soc. Aster. If Do, Store. Soc. Aster. S	t Bo, Kore. Bo,	If ico, store. If ico, store. If ico, store. If ico, store. Send (a,b) lift. Repeat. If ico, store. If ico, store.	-		Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$
Is a - box, smooth? If so, store. Send (a, b) right. Repeat.	signal and the second s	$ \begin{array}{c c} \mathbf{x}_{a} = \frac{h_{1}}{2} & \mathbf{x}_{a} = \frac{h_{2}}{2} & \mathbf{x}_{a} = \frac{h_{1}}{2} & \mathbf{x}_{a} = $	$\label{eq:starsest} \begin{array}{c c} is a - b v_1 & is a - b v_1 \\ is a - b v_1 \\ is a - b v_1 \\ is a - $	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$			smooth?	smooth?	smooth?	smooth?
is a - bosy smooth? If so, store. Send (a, b) up. Repeat. Is a - bosy	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c} \text{is } a - b \alpha_1 \\ \text{smooth} \\ \text{th } a , \text{ tore,} \\ \text{th } a , \text{tore,} \\ \text{th } a - b \alpha_1 \\ $	$ \begin{array}{c} \ \hat{u} - b v_{1} \\ \text{second} \\ \ \hat{v}_{n} \text{ cond} \\ \ \\ \ \\ \text{second} \\ \ \\ \ \\ \text{second} \\ \ \\ \ \\ \ \\ \ \\ \ \\ \ \\ \ \\ \ \\ \ \\ $			If so, store.	If so, store.	If so, store.	If so, store.
smpoch? If 6, store. Send [5] Is a - bo ₁₃ smpoch?	smooth? If so, know, if so, know, is a - brux, smooth? is a - brux, smooth?	$\begin{array}{c c} \operatorname{second}^{n} & \operatorname{second}^{n} \\ \operatorname{If} \end{transformation} \\ If$	smooth W to dear st. Send (A. Song Ching Channel st. Send (A. Song Channel st. Send	$\begin{array}{c} \begin{array}{c} \text{smooth}^{2} \\ \text{from only}^{2} \\ \text{south}^{2} \\ so$			Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
If so, store. Send (<i>a</i> , <i>b</i>) up. Repeat.	Seed a property of the second	H So. 15kp. Send (, A) up. Repeat. H So. 1000 Repeat. H So. 1000 Repeat.	A Serd (a) bit. Serd (a bit) R. Serd (a) bit. Serd (a bit) Reput Reput	Wiss store Sind (x, k) sht Wiss store Wiss store Send (x, k) sht Sind (x, k) sht Wiss store Wiss store Repeat Property Wiss store Repeat			Repeat.	Repeat.	Repeat.	Repeat.
Is a - box; smooth? If so, store. Send (a, b) right. Repeat.	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	$ \begin{array}{ c c c c c c } \hline h & a & Bn_1 & & h & a & -bn_2 & & & h & a & -bn_1 & & & h & a & -bn_1 & & & \\ \hline & mooth^1 & mooth^1 & & & & & & & \\ \hline & mooth^1 & & & & & & & \\ \hline & & & & & & & & \\ \hline & & & &$	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$			Is $a - b \alpha_9$	Is $a - b\alpha_{10}$	Is $a - b \alpha_{11}$	Is $a - b\alpha_{12}$
Is a - bos smooth? If so, store. Send (a, b) up. Repeat. Is a - bos	$\begin{array}{c} \mathbf{k} = -b \mathbf{r}_{1} \\ \mathbf{s} = b \mathbf{r}_{2} \\ \mathbf{s} = b \mathbf{r}_{1} \\ \mathbf{s} = b \mathbf{r}_{2} \\ \mathbf{s} = b \mathbf{r}_{1} \\ \mathbf{s} = b \mathbf{r}_{2} $	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	$ \begin{array}{c} \text{is } a - b \alpha_{7} \\ \text{smooth}^{7} \\ \text{if } \alpha_{6} \\ \text{tors} \\ \text{smooth}^{7} \\ smooth$	$ \begin{array}{c} \begin{matrix} \mathbf{k} a - b \mathbf{x}_1 \\ \mathbf{s} = b \mathbf{x}_1 \\ $			smooth?	smooth?	smooth?	smooth?
semocth? If so, store. Send (a, b) right Repeat. Is $a - br_{13}$ semocth?	smooth? smooth? smooth? smooth? smooth? smooth? simoth? simoth	$\begin{array}{c} \operatorname{smooth}^{1} & \operatorname{smooth}^{2} \\ \operatorname{smooth}^{1} & \operatorname{smooth}^{2} \\ \operatorname{smooth}^{2} \\ \operatorname{smooth}^{2} & \operatorname{smooth}^{2} \\ smoot$	a smooth? We know. at. Sind (a,b) right. Sind (a,b) right. sub a - bros. sub	smooth? smooth? <t< th=""><th></th><th></th><th>If so, store.</th><th>If so, store.</th><th>If so, store.</th><th>If so, store.</th></t<>			If so, store.	If so, store.	If so, store.	If so, store.
Send		A - By Wash Wash See (A a) Report					() =	Send (<i>a</i> , <i>b</i>) right.	() =	Send (<i>a</i> , <i>b</i>) down.
	Repeat.	Repeat.	Repeat.	Repeat.			Repeat.	Repeat.	Repeat.	Repeat.
	s $a - b \alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$			$ls a - b\alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$
	smooth?	smooth?	smooth?	smooth?	$\overline{}$		smooth?	smooth?	smooth?	smooth?
	f so, store.	If so, store.	If so, store.	If so, store.		\frown	\setminus If so, store.	If so, store.	If so, store.	If so, store
Ser	nd (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.			\bigwedge nd (a, b) up.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>)/
	Repeat.	Repeat.	Repeat.	Repeat.			Repeat.	Repeat.	Repeat.	Repe
										$\overline{}$
	_		-							
	Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$				$b\alpha_2$	ls a – ba	
	smooth?	smooth?	smooth?	smooth?						

IS $a - b\alpha_1$	Is $a - b\alpha_2$	IS $a - b\alpha_3$	IS $a - b\alpha_4$	
smooth?	smooth?	smooth?	smooth?	
If so, store.	If so, store.	If so, store.	If so, store.	
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.	
Repeat.	Repeat.	Repeat.	Repeat.	
Is $a - b\alpha_5$	Is $a - b \alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$	
smooth?	smooth?	smooth?	smooth?	
If so, store.	If so, store.	If so, store.	If so, store.	
Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	
Repeat.	Repeat.	Repeat.	Repeat.	
Is $a - b\alpha_9$	Is $a - b lpha_{10}$	Is $a - b lpha_{11}$	Is $a - b\alpha_{12}$	
smooth?	smooth?	smooth?	smooth?	
16	16+	16+	16+	ı ∎'

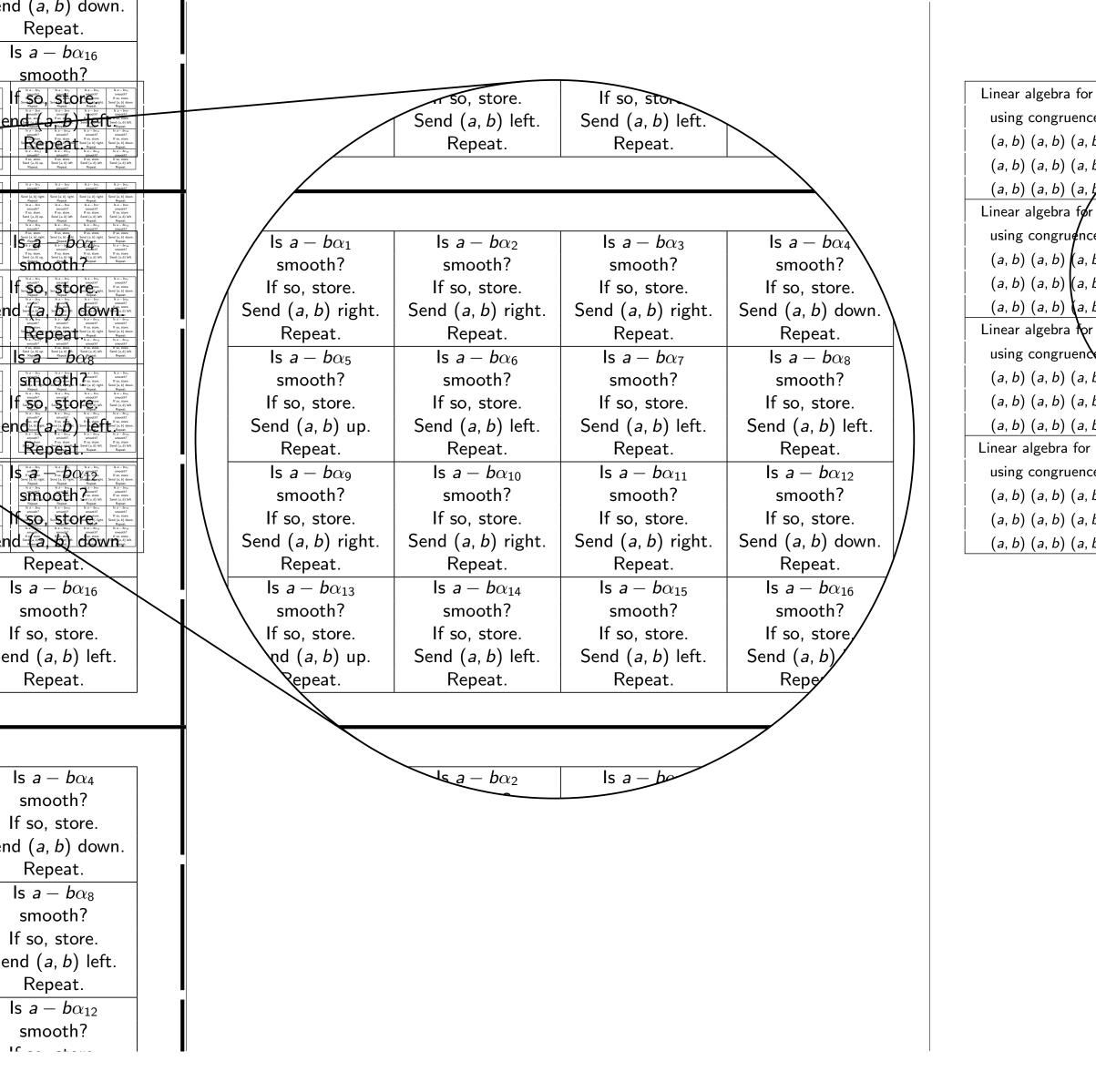
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.					
Repeat.	Repeat.	Repeat.					
Is $a - b\alpha_{14}$	Is $a - b \alpha_{15}$	Is $\textit{a} - \textit{b} lpha_{16}$					
smooth?	smooth?	smooth?	-				
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c} is \ a - b \alpha_1 \\ model, \\ mod$	$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$				so, store.	If so, stor
u → No toot to	$k_{a} = b_{01}$ $k_{a} = b_{01}$	k = - box smooth if so at Smooth if so at i				Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	и подоб) ⁷ и подоб) ⁷ и под	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$				Repeat.	Repeat.
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\label{eq:constraint} \begin{array}{ c c c c c c } \hline & & & & & & & & & & & & & & & & & & $	smooth/ Ho, store. Wo, store. <th< td=""><td>_</td><td></td><td></td><td></td><td></td></th<>	_				
$a = bn_1$ $ba = bn_2$ $ba = bn_1$	i is a - br ₁ is a - br ₂ is a - br ₃	Is $a - bx_1$ smooth? smooth? smooth? smooth? smooth? smooth? Sand (a, b) fourn. Sand (a, b) fourn. Sand (b, b) fourn. Sand (b, c) fourn. Sand (b, c) fourn. Sand (b, c) fourn.					
$\label{eq:constraint} \begin{array}{cccc} \operatorname{respect.} & \operatorname{respect.} \\ \operatorname{source} & \operatorname{source} & \operatorname{respect.} \\ \operatorname{source} & \operatorname{source} & \operatorname{respect.} \\ \operatorname{source} & \operatorname{source} & \operatorname{respect.} \\ $	$\label{eq:constraint} \begin{array}{cccc} compared. comp$	$\label{eq:constraint} \begin{array}{c} \operatorname{respect.} & \operatorname{respect.} & \operatorname{respect.} & \operatorname{respect.} & \operatorname{respect.} \\ \operatorname{Is a - box} & \operatorname{seco-box} & seco-$					
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	10 3 B → Drig unceDT 10 → Drig U	$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$			Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$
$\begin{array}{ccc} \operatorname{model}^{1} & \operatorname{model}^{1}$	It is noted if it is				smooth?	smooth?	smooth?
	1 5 or (a b) of (b) - Repeat.				If so, store.	If so, store.	If so, store.
$a - ba_{11}$ is $a - ba_{22}$ is $a - ba_{23}$ is $a - ba_{24}$ is $a - ba_{24}$	n service (service) are able to a service and the service are able to a service are able t	$ s a - bc_0$		/	Send (a, b) right.	Send (a, b) right.	Send (a, b) right.
Baped Construction	It is a - bin the	IS 2 - 20122 IS 2 - 2012 IS 2 - 20121 IS 2 - 20122	İ	/	Repeat.	Repeat.	Repeat.
Open Arrow North Arrow	$\begin{array}{c c c c c c c c c c c c c c c c c c c $			/	Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$
			Η		smooth?	smooth?	smooth?
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Is a box Is a box Is a box Is a box	Repeat. Repeat. Repeat. Repeat. Repeat.			If so, store.	If so, store.	If so, store.
$a = b \alpha_0$ is $a = b \alpha_{10}$ is $a = b \alpha_{11}$ is $a = b \alpha_{12}$ is $a = b \alpha_0$ is $a = b \alpha_0$	C Series (as) or (as) (as) (b) (b) (b) (b) (b) (b) (b) (b) (b) (b	Is $a - b\alpha_{12}$ Is $a - b\alpha_{0}$ Is $a - b\alpha_{10}$ Is $a - b\alpha_{11}$ Is $a - b\alpha_{12}$			Send (a, b) up.	Send (a, b) left.	Send (a, b) left.
	полос 1 лле. полос				Repeat.	Repeat.	Repeat.
					Is $a - b\alpha_9$	Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$
is a chen If so, store. If so, store If so,	nt Sond Londong Sondard Sondar	U do, korec. Star (a, b) down Star (a, b) down			smooth?	smooth?	smooth?
					If so, store.	If so, store.	If so, store.
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	A substitution of the subs	Same (a, b) come. Same (a, b) come. Same (a, b) come. Same (a, b) come. Repaid. Repaid. Same (a, b) come. Same (a, b) come. Same (a, b) come. Is a - briss. Same (a, b) come. Same (a, b) come. Same (a, b) come. Same (a, b) come. If a - briss. Same (a, b) come. If (b, a) come. If (b, a) come. If (b, a) come. Same (a, b) come. Same (a, b) come. If (b, a) come. If (b, a) come. If (b, a) come.			Send (a, b) right.	Send (a, b) right.	Send (a, b) right.
Repeat.	Repeat.	Repeat.			Repeat.	Repeat.	Repeat.
Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$			$\sqrt{\frac{1}{1} \text{ Is } a - b\alpha_{13}}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$
smooth?	smooth?	smooth?			smooth?	smooth?	smooth?
If so, store.	If so, store.	If so, store.			If so, store.	If so, store.	If so, store.
Send (a, b) left.	Send (a, b) left.	Send (a, b) left.			nd (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (a, b) left.
Repeat.	Repeat.	Repeat.			epeat.	Repeat.	Repeat.
Tepeat.	Repeat.	Переаг.			lepear.	Repeat.	Repeat.
Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$				$bar{a} - b\alpha_2$	Is a – bo
smooth?	smooth?	smooth?					
If so, store.	If so, store.	If so, store.					
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) down.					
Repeat.	Repeat.	Repeat.					
Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$					
smooth?	smooth?	smooth?					
If so, store.	If so, store.	If so, store.					
Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.	Send (a, b) left.					
Repeat.	Repeat.	Repeat.					
Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$	Is $a - b\alpha_{12}$					
smooth?	smooth?	smooth?					
		16		■ 1			



Linear algebra for N_1 using congruences (a, b) (a, b) (a, b)(a, b) (a, b) (a, b)(a, b) (a, b) (a, b) Linear algebra for N_5 using congruences (a, b) Linear algebra for N_9 using congruences (a, b) inear algebra for N_{12} using congruences (a, b) (a, b) (a, b)(a, b) (a, b) (a, b)(a, b) (a, b) (a, b)

Send (a, b) right.	Send (a, b) down.						
Repeat.	Repeat.						
Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$						
smooth?	smooth?						
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{bmatrix} h & a - bn_1 \\ monoth \\ strong Cont} \\ strong C$				so, store.	If so, stor	
	Impart. Impart. <thimpart.< th=""> <th< td=""><td></td><td></td><td>/</td><td>Send (<i>a</i>, <i>b</i>) left.</td><td>Send (a, b) left.</td><td></td></th<></thimpart.<>			/	Send (<i>a</i> , <i>b</i>) left.	Send (a, b) left.	
It a - born in contra in contra	$\begin{array}{c} \mathbf{x}_{a} - b\mathbf{x}_{a} \\ \mathrm{smooth}^{T} \\ \mathbf{y}_{not} \operatorname{store}_{t} \\ \mathrm{See}(a,b) \operatorname{down}_{t} \\ \mathrm{See}(a,b) \operatorname{See}(a,b) $				Repeat.	Repeat.	
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	smooth? smooth? smooth? smooth?				•	•	
Is a - br ₁ mosth? smooth? smooth? smooth?	Is a - bro; Is a - bro; Is a - bro; Is a - bro; smooth? smooth? smooth? smooth?						
Sind (a, b) right. Sind (a	Same(a, b) down. Same(a, b) right.						\backslash
$\label{eq:result} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	Repeat. Repeat. <t< td=""><td></td><td></td><td></td><td>le a bar</td><td>la a hou</td><td></td></t<>				le a bar	la a hou	
Ind (a) ingt: Sea (a)	Image (L) Image (L) <t< td=""><td></td><td></td><td>$\int \mathbf{s} \mathbf{a} - \mathbf{b} \alpha_1$</td><td>Is $a - b\alpha_2$</td><td>Is $a - b\alpha_3$</td><td>$s a - b\alpha_4 \rangle$</td></t<>			$\int \mathbf{s} \mathbf{a} - \mathbf{b} \alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	$ s a - b\alpha_4 \rangle$
	br-br moth brind bri			smooth?	smooth?	smooth?	smooth?
$ s _a - b\alpha_b$	$ s a - b\alpha_k$			If so, store.	If so, store.	If so, store.	If so, store.
	-Send - (2, 2, 3) have been been been been been been been be			Send (a, b) right.	Send (a, b) right.	Send (a, b) right.	Send (a, b) down.
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	U to, store, Regards Set (a, b) con- text U to, store, Set (a, b) con- text U to, sto			Repeat.	Repeat.	Repeat.	Repeat.
incontrol in analysis incontrol in analysis in an intervention of the second in interventintervention of th				Is $a - b\alpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$
repeat. repeat. repeat.	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$			smooth?	smooth?	smooth?	smooth?
	$ \begin{array}{c} \begin{array}{c} \begin{array}{c} \mbox{array}{ll} \\ \mbox{array}{l$			If so, store.	If so, store.	If so, store.	If so, store.
	Install in the second s			Send (<i>a</i> , <i>b</i>) up.	Send (a, b) left.	Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.
Weinkern, Wisser Sind (A) Mith. Sond (A) A BR PORT Sind (A) Mith. Sond (A) Mith. Repart Repart PORT Sind (A) Mith. Repart.				Repeat.	Repeat.	Repeat.	Repeat.
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	television and televi			Is $a - b\alpha_9$	Is $a - b \alpha_{10}$	Is $a - b\alpha_{11}$	Is $a - b\alpha_{12}$
lea house lea house lea house lea house lea house	$ \begin{array}{c c} h_{A} = b v_{1} \\ model \\ mo$			smooth?	smooth?	smooth?	smooth?
If so, store, $ \begin{array}{c c} \text{If so, store,} \\ \text{Single L}, \text{b} \text{ of ght,} \\ \text{Repart,} \\ \text{Repart,} \\ \text{If } a - b n_{11} \\ \text{smooth}^{T} \\ \text{smooth}^{T} \\ \end{array} \begin{array}{c c} \text{Single L}, \text{b} \text{ of ght,} \\ \text{Single L}, \text{c} \text{ of ght,} \\ \text{Single L}, \\ \text{Single L}, \text{c} \text{ of ght,} \\ \text{Single L}, \\ \text{Single L}, \text{c} \text{ of ght,} \\ \text{Single L}, \\ \\ \text{Single L}, \\ \text{Single L}, \\ \\ Sin$	Image: State of the state o			If so, store.	If so, store.	If so, store.	If so, store.
				Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
Repeat.	Repeat.			Repeat.	Repeat.	Repeat.	Repeat.
Is $a - b lpha_{15}$	Is $a - b \alpha_{16}$			Is $a - b\alpha_{13}$	Is ${\it a}-{\it b}lpha_{14}$	Is $a - b \alpha_{15}$	Is $a - b\alpha_{16}$
smooth?	smooth?	\searrow		\setminus smooth?	smooth?	smooth?	smooth? /
If so, store.	If so, store.		\searrow	\setminus If so, store.	If so, store.	If so, store.	If so, store
Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.			\checkmark nd (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>)
Repeat.	Repeat.			Repeat.	Repeat.	Repeat.	Repe
Is $a - b\alpha_3$	Is $a - b\alpha_4$				$b\alpha_2 - b\alpha_2$	Is a – ba	
smooth?	smooth?						
If so, store.	If so, store.						
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.						
Repeat.	Repeat.						
Is $a - b\alpha_7$	Is $a - b\alpha_8$						
smooth?	smooth?						
If so, store.	If so, store.						
Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.						
Repeat.	Repeat.						
Is $a - b\alpha_{11}$	Is $a - b\alpha_{12}$						
smooth?	smooth?						

-
Linear algebra for N_2
using congruences
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
Linear algebra for N_6
using congruences
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
Linear algebra for N_{10}
using congruences
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
Linear algebra for N_{14}
using congruences
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)



· N ₁	Linear algebra for N_2	Linear algebra for N_3	L
ces	using congruences	using congruences	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
þ	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
N ₅	Linear algebra for N_6	Linear algebra for N_7	L
ces	using congruences	using congruences	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
· N ₉	Linear algebra for N_{10}	Linear algebra for N_{11}	L
es	using congruences	using congruences	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
<i>N</i> ₁₃	Linear algebra for N_{14}	Linear algebra for N_{15}	L
ces	using congruences	using congruences	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
<i>b</i>)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
			<u> </u>

		so, store.	If so, stor	
		Send (a, b) left.	Send (<i>a</i> , <i>b</i>) left.	\searrow
		Repeat.	Repeat.	
	Is $a - b\alpha_1$	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b\alpha_4$
	smooth?	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store.	If so, store.
	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (a, b) down.
	Repeat.	Repeat.	Repeat.	Repeat.
	Is $a - b lpha_5$	Is $a - b\alpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$
	smooth?	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store.	If so, store.
	Send (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
	Repeat.	Repeat.	Repeat.	Repeat.
	Is $a - b lpha_9$	Is $a - b\alpha_{10}$	Is $a - b\alpha_{11}$	Is $a - b\alpha_{12}$
	smooth?	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store.	If so, store.
	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
	Repeat.	Repeat.	Repeat.	Repeat.
N	Is $a - b\alpha_{13}$	Is $a - b\alpha_{14}$	Is $a - b\alpha_{15}$	Is $a - b\alpha_{16}$
	<pre>smooth?</pre>	smooth?	smooth?	smooth? /
	\setminus If so, store.	If so, store.	If so, store.	If so, store
	\checkmark nd (<i>a</i> , <i>b</i>) up.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>)/
	Repeat.	Repeat.	Repeat.	Repe
		<u></u>	/	
		$b\alpha = b\alpha_2$	ls a – bo	

Linear algebra for N_1	Linear algebra for N_2	Linear algebra for N_3	Linear algebra for N_4
using congruences	using congruences	using congruences	using congruences
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
Linear algebra for N_5	Linear algebra for N_6	Linear algebra for N_7	Linear algebra for N_8
using congruences	using congruences	using congruences	using congruences
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
Linear algebra for N_9	Linear algebra for N_{10}	Linear algebra for N_{11}	Linear algebra for N_{12}
using congruences	using congruences	using congruences	using congruences
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
Linear algebra for N_{13}	Linear algebra for N_{14}	Linear algebra for N_{15}	Linear algebra for N_{16}
using congruences	using congruences	using congruences	using congruences
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)

	so, store.	If so, stor	
	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	
	Repeat.	Repeat.	
		-	
	Is $a - b\alpha_2$	Is $a - b\alpha_3$	Is $a - b \alpha_4$
	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store. 🔪
:.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
	Repeat.	Repeat.	Repeat.
	Is $a - b lpha_6$	Is $a - b\alpha_7$	Is $a - b\alpha_8$
	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store.
	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
	Repeat.	Repeat.	Repeat.
	Is $a - b lpha_{10}$	Is $a - b\alpha_{11}$	Is $a - b \alpha_{12}$
	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store.
	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
	Repeat.	Repeat.	Repeat.
	Is $a - b lpha_{14}$	Is $a - b \alpha_{15}$	Is $a - b\alpha_{16}$
	smooth?	smooth?	smooth?
	If so, store.	If so, store.	If so, store
	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>)
	Repeat.	Repeat.	Repe

		i	
Linear algebra for N_1	Linear algebra for N_2	Linear algebra for N_3	
using congruences	using congruences	using congruences	┝
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
Linear algebra for N_5	Linear algebra for N_6	Linear algebra for N_7	
using congruences	using congruences	using congruences	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a b)	Ī
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	Ļ
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
Linear algebra for N_9	Linear algebra for N_{10}	Linear algebra for N_{11}	
using congruences	using congruences	using congruences	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a b)	
Linear algebra for N_{13}	Linear algebra for N_{14}	Linear algebra for N_{15}	
using congruences	using congruences	using congruences	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	
(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	

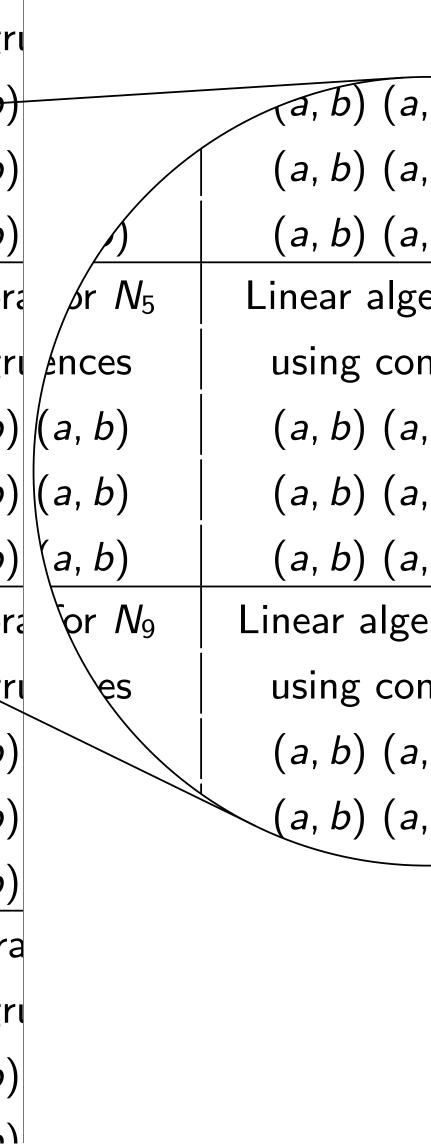
 $|\mathbf{s} \mathbf{a} - \mathbf{b} \alpha_2|$ Is $\mathbf{a} - \mathbf{b} \alpha_2$

using congri inear algebra for NA (a, b) (a, b) (a, b)(a, b) (a, b) (a, b) (a, b)(a, b) (a, b) (a, b) (a, b)(a, b) (a, b) (aLinear algebra for N_{s} by using congruences bLinearb) (a, b) (a, b) Linearb) a gebra $\delta r N_5$ (a, b) (a, b) (a, b) Linear algebra for N12 USING CONGULATION *k*nces (a, b) $\begin{pmatrix} \text{Liftear algebra for} \\ \textbf{a} \\ \textbf{strig congruen} \\ \textbf{es} \\ \textbf{a}, b \\ \textbf{b} \\ \textbf{b} \\ \textbf{b} \\ \textbf{congruen} \\ \textbf{co$ (a, b) a, b) $\nabla r N_9$ Linear algebra using congri ęs (a, b) (a, b)(a, b) (a, b)(a, b) (a, b)Linear algebra using congri (a, b) (a, b)(2h)(2h)

using congri

If so, stor	
Send (<i>a</i> , <i>b</i>) left.	
Repeat.	
	\longrightarrow
	\backslash
Is $a - b\alpha_3$	Is $a - b\alpha_4$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
Repeat.	Repeat.
Is $a - b\alpha_7$	Is $a - b \alpha_8$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>) left.
Repeat.	Repeat.
Is $a - b lpha_{11}$	Is $\mathit{a} - \mathit{b} lpha_{12}$
smooth?	smooth?
If so, store.	If so, store.
Send (<i>a</i> , <i>b</i>) right.	Send (<i>a</i> , <i>b</i>) down.
Repeat.	Repeat.
Is $a - b lpha_{15}$	Is $a - b\alpha_{16}$
smooth?	smooth?
If so, store.	If so, store
Send (<i>a</i> , <i>b</i>) left.	Send (<i>a</i> , <i>b</i>)/
Repeat.	Repe
ls a – ba	

			1				
ſ	Linear algebra for N_1	Linear algebra for N_2	Linear algebra for	r <i>N</i> 3 L	near algebra f	$\sim N_{4}$	b
	using congruences	using congruences	using congruent	es	using congrue	ndes ,	D)
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a	b)	(a, b) (a, b) (a	a, b)	. \
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a	b)	(a a), (b)(a	a,(b) a ,	b)
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a	<i>b</i>)	(a, b) (a, b) (a)
	Linear algebra for N_5	Linear algebra for N_6	Linear algebra for		inear algebra f	$\sim N_{\rm s}$	b)
ļ	using congruences	using congruences	using congruen	ces	using congrue	ndesa,	b)
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a		(a, b) (a, b) (a		
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	b)		alge	br
ļ	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a	<i>b</i>)	(a, b) (a, b) (a	a, b)	
ļ	Linear algebra for N_9	Linear algebra for N_{10}	Linear algebra for	<i>N</i> ₁₁ L	inear algebra fo USING using congouei		σr
ļ	using congruences	using congruences	using congruence	ces	using congrue	nces	gr
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	<i>b</i>)	(a, b) (a, b) (a	a, b)	<i>,</i> \
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	<i>b</i>)	(a a), (b)(a	a,(b) a ,	b)
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a	<i>b</i>)	(a, b) (a, b) (a	a, b)	/
ļ	Linear algebra for N_{13}	Linear algebra for N_{14}	Linear algebra for	<i>N</i> ₁₅ L	(a, b) (a, b) (a)	rN_{16}	b)
ļ	using congruences	using congruences	using congruen	ces	ising congrue	ndes ,	D)
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	<i>b</i>)	(a, b) (a, b) (a	a, b)	. \
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	<i>b</i>)	(@), (D)(2	a,(b) Ə , [b)
l	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a		(a, b) (a, b) (a, b)	(b)	/
					near a	loe	hr
						inge	
					using		ar
					using	CON	8í
					(2b)	()	と)
					(a, b)	(a,	D
						1	1)
					(a, b)	(<i>a</i> ,	b)
			i			-	-
					(a, b)	(<i>a</i> ,	b)
						('	/
				l i	near a	lge	ora
						.0	••••
					using	con	σr
					45115	CON	6'
					(<i>a</i> , <i>b</i>)	()	と)
					(a, b)	(a,	\mathcal{D}
					(2 h)	(-	しく
			I				r 1



			1		0		
				using	congri		
	Linear algebra for N_1	Linear algebra for N_2	Linear algebra for	$r N_3$ Linear algebra	$fo(\frac{N_4}{a}, b)$		
	using congruences	using congruences	using congruent	es using congrue	endes, D)		
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a				
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a				
	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,				1
	Linear algebra for N_5	Linear algebra for N_6	Linear algebra for		$a_{a}^{\text{for}}(a, b)$		
Is $a - b\alpha_4$	using congruences	using congruences	using congruenc	i			
smooth?	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,	-			
If so, store. \setminus	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,		algebra	$ ho$ r N_5	
Send (a, b) down.	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a				İ
Repeat.	Linear algebra for N_9	Linear algebra for N_{10} .	Linear algebra for		congriged and the second secon	<i>k</i> nces	
Is $a - b\alpha_8$	using congruences	using congruences	using congruenc	i	i		
smooth?	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,		(a, b)	(a b)	
If so, store.	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a,			(a, b)	
Send (a, b) left.	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)				Ì
$\frac{\text{Repeat.}}{\text{Is } a - b\alpha_{12}}$	Linear algebra for N ₁₃ using congruences	Linear algebra for N_{14} using congruences	Linear algebra for using congruenc	$(A \cap A)$	$a_{a}(a_{a}, b)$	(<i>a</i> , <i>b</i>)	
smooth?	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)	(a, b) (a, b) (a		i		1
If so, store.	(a, b) (a, b) (a, b) (a, b) (a, b) (a, b)	(a, b) (a, b) (a, b) (a, b) (a, b) (a, b)	(a, b) (a, b) (a, b) (a, c)			(a,b)	
Send (a, b) down.	(a, b) (a, b) (a, b) (a, b) (a, b) (a, b)	(a, b) (a, b) (a, b) (a, b) (a, b) (a, b)	(a, b) (a, b)			(a, b)	
Repeat.	(a, b) (a, b) (a, b)	(a, b) (a, b) (a, b)					
Is $a - b\alpha_{16}$				Linear	algebra	for N ₉	
smooth?			i				1
If so, store				using	congri	es	
Send (<i>a</i> , <i>b</i>)				using	const		
Repe				(-1)	(-1)		
				(<i>a</i> , <i>b</i>)	(a, b)		
$\overline{}$							Ľ
				(a,b)	(a, b)		
				(4, 5)	(4,2)		
				(a,b)	(a, b)		
				(a, D)	(a, b)		
			Ī				
				Linear a	algebra		
				using	congri		
				using	Congri		
				(a, b)	(<i>a</i> , <i>b</i>)		
				(a b)	(2h)		

