

Analyzing the Quadratic Sieve: An Undergraduate Research Case-history



1/59



Analyzing the Quadratic Sieve: An Undergraduate Research Case-history

or

The Tails of Two Sieves

Neil J. Calkin

Clemson University

MAA Short Course

San Antonio

January 10, 2006.



1/59

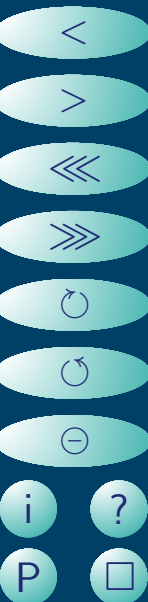




The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.





The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.

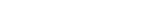




The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.
- Experiments as tool for learning about the nature of a problem, generating conjectures, new ideas.





The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.
- Experiments as tool for learning about the nature of a problem, generating conjectures, new ideas.
- Fallibilism in action: let's make mistakes too!





The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.
- Experiments as tool for learning about the nature of a problem, generating conjectures, new ideas.
- Fallibilism in action: let's make mistakes too!
- Experimentation as a goal itself.





The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.
- Experiments as tool for learning about the nature of a problem, generating conjectures, new ideas.
- Fallibilism in action: let's make mistakes too!
- Experimentation as a goal itself.
- Work in progress: getting undergraduates to write up the work can slow things down!





The Clemson REU 2004

The Clemson Research Experience for Undergraduates in Computation Number Theory and Combinatorics

- Computational focus from/at beginning.
- Experiments as tool for getting started.
- Experiments as tool for learning about the nature of a problem, generating conjectures, new ideas.
- Fallibilism in action: let's make mistakes too!
- Experimentation as a goal itself.
- Work in progress: getting undergraduates to write up the work can slow things down!
Most of all: learning about research by doing research.





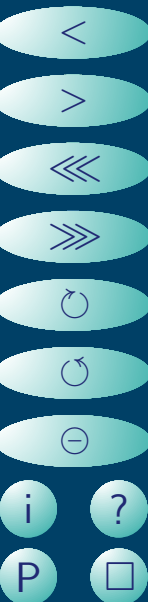
Co-conspirators

- Kevin James, Clemson University, co-supervisor of the REU
- Tim Flowers, Clemson University, graduate student assisting the REU
- Shannon Purvis, Clemson University, graduate student assisting the REU
- Kim Bowman, Clemson University, undergraduate REU participant (2004)
- Zach Cochran, University of Georgia, undergraduate REU participant (2004)
- Katie Field, William and Mary, undergraduate REU participant (2005)
- Tina Little, William and Mary, undergraduate REU participant (2005)
- Ryan Witko, New York University, undergraduate REU participant (2005)



Factorization algorithms

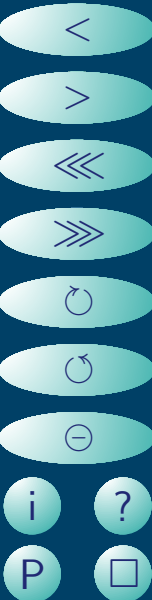
- Trial division — the sieve of Eratosthenes





Factorization algorithms

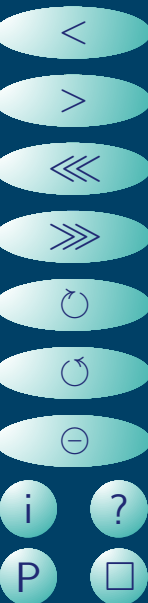
- Trial division — the sieve of Eratosthenes
- Fermat's algorithm





Factorization algorithms

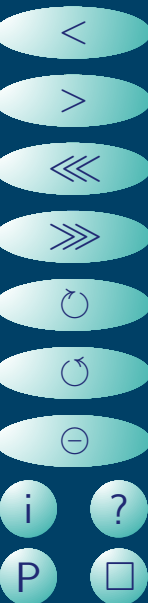
- Trial division — the sieve of Eratosthenes
- Fermat's algorithm
- Pomerance's Quadratic Sieve
- The Number Field Sieve





Trial Division

- Works well to factor n if and only if n has a small prime divisor
- Can take on the order of $n^{1/2}/\log n$ attempts to find a factor





Fermat's Algorithm

- Look at the squares b^2 which are bigger than n
- If $b^2 - n = a^2$ is a square, then $n = b^2 - a^2 = (b + a)(b - a)$
- Conversely, if $n = xy$ then set $b = \frac{x+y}{2}$, $a = \frac{x-y}{2}$
- (Doesn't work if x, y have opposite parity, but then n is even!)





Fermat's Algorithm

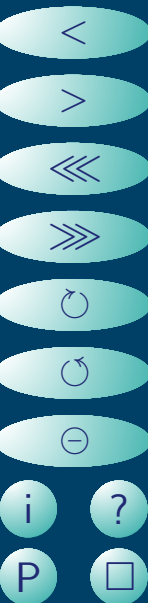
- Look at the squares b^2 which are bigger than n
- If $b^2 - n = a^2$ is a square, then $n = b^2 - a^2 = (b + a)(b - a)$
- Conversely, if $n = xy$ then set $b = \frac{x+y}{2}$, $a = \frac{x-y}{2}$
- (Doesn't work if x, y have opposite parity, but then n is even!)
- Big problem with the method: how to find b
- Works well if n has a factorization into parts which are very close
- May need to examine on the order of \sqrt{n} b 's to find a factorization of n .





Kraitchik's Method

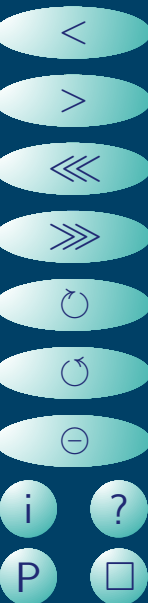
- Find two squares u^2, v^2 so that $n|v^2 - u^2$
- Then $(v - u)(v + u) \equiv 0 \pmod{n}$
- If $u \not\equiv \pm v \pmod{n}$ then this gives a non-trivial factorization





Kraitchik's Method

- Find two squares u^2, v^2 so that $n|v^2 - u^2$
- Then $(v - u)(v + u) \equiv 0 \pmod{n}$
- If $u \not\equiv \pm v \pmod{n}$ then this gives a non-trivial factorization
- Big problem with the method: how to find u, v : not really an algorithm as such
- Shows that factoring n is really no harder than taking square roots mod n .

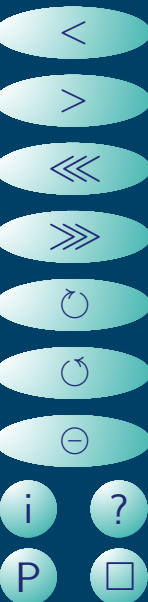


Kraitchik, Dixon:

- Create a set of small primes \mathcal{B}
- Find values x_i so that $f(x_i) = x_i^2 \pmod{n}$ is small, and factorable using only the primes in \mathcal{B} .



8/59



Kraitchik, Dixon:

- Create a set of small primes \mathcal{B}
- Find values x_i so that $f(x_i) = x_i^2 \pmod{n}$ is small, and factorable using only the primes in \mathcal{B} .

Call such an $f(x_i)$ *\mathcal{B} -smooth*



8/59



Kraitchik, Dixon:

- Create a set of small primes \mathcal{B}
- Find values x_i so that $f(x_i) = x_i^2 \pmod{n}$ is small, and factorable using only the primes in \mathcal{B} .

Call such an $f(x_i)$ *\mathcal{B} -smooth*

- Find a subset of the x_i 's (for which $f(x_i)$ is \mathcal{B} -smooth) for which the product of the $f(x_i)$ is a square
- Then for this subset of values, $\prod x_{i_j}^2 \equiv \prod f(x_{i_j}) \pmod{n}$ and both sides are squares



- Create a set of small primes \mathcal{B}
- Find values x_i so that $f(x_i) = x_i^2 \pmod{n}$ is small, and factorable using only the primes in \mathcal{B} .

Call such an $f(x_i)$ \mathcal{B} -smooth

- Find a subset of the x_i 's (for which $f(x_i)$ is \mathcal{B} -smooth) for which the product of the $f(x_i)$ is a square
- Then for this subset of values, $\prod x_{i_j}^2 \equiv \prod f(x_{i_j}) \pmod{n}$ and both sides are squares
- Note that since the $f(x_{i_j})$ all factor over the factor base \mathcal{B} we actually know a square root of $\prod f(x_{i_j})$



Main problems:

- How big a factor base do we need?



9/59



Main problems:

- How big a factor base do we need?
- How do we find suitable x_i ? How many x_i do we need to consider? Which ones?



9/59



Main problems:

- How big a factor base do we need?
- How do we find suitable x_i ? How many x_i do we need to consider? Which ones?
- How to find a subset of the $f(x_{i_j})$ whose product is a square?

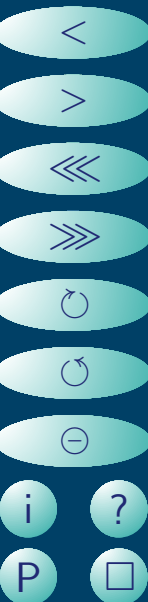




The Quadratic Sieve

Pomerance's brilliant observation: if we are considering values x just larger than \sqrt{n} , then $f(x) = x^2 - n$ and we can use a sieve!

- $p|f(x)$ if and only if $p|f(x + p)$





The Quadratic Sieve

Pomerance's brilliant observation: if we are considering values x just larger than \sqrt{n} , then $f(x) = x^2 - n$ and we can use a sieve!

- $p|f(x)$ if and only if $p|f(x+p)$
(provided $n < x^2 < 2n$)





The Quadratic Sieve

Pomerance's brilliant observation: if we are considering values x just larger than \sqrt{n} , then $f(x) = x^2 - n$ and we can use a sieve!

- $p|f(x)$ if and only if $p|f(x+p)$
(provided $n < x^2 < 2n$)
- Consider the values $x_i = \lfloor \sqrt{n} \rfloor + i$
- Find all \mathcal{B} -smooth values x_{i_j}





The Quadratic Sieve

Pomerance's brilliant observation: if we are considering values x just larger than \sqrt{n} , then $f(x) = x^2 - n$ and we can use a sieve!

- $p|f(x)$ if and only if $p|f(x+p)$
(provided $n < x^2 < 2n$)
- Consider the values $x_i = \lfloor \sqrt{n} \rfloor + i$
- Find all \mathcal{B} -smooth values x_{i_j}
(those x_{i_j} for which $f(x_{i_j})$ factors completely over the factor base \mathcal{B}).





The Quadratic Sieve

Pomerance's brilliant observation: if we are considering values x just larger than \sqrt{n} , then $f(x) = x^2 - n$ and we can use a sieve!

- $p|f(x)$ if and only if $p|f(x + p)$
(provided $n < x^2 < 2n$)
- Consider the values $x_i = \lfloor \sqrt{n} \rfloor + i$
- Find all \mathcal{B} -smooth values x_{i_j}
(those x_{i_j} for which $f(x_{i_j})$ factors completely over the factor base \mathcal{B}).
- Factorization of the $f(x_i)$ is performed using (essentially) a sieve of Eratosthenes restricted to the factor base.

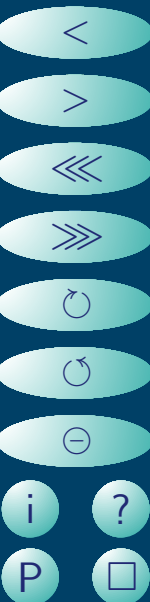


How to find a subset of the x_i for which the product of $f(x_{i_j})$ is a square?

- Start with a factor base \mathcal{B} of size k
- For each \mathcal{B} -smooth $f(x_i)$, create a binary exponent vector $\underline{v}(x_i) \in GF_2^k$ representing the parity of the powers of $p \in \mathcal{B}$ which divide $f(x_i)$.
- Then a subset x_{i_1}, \dots, x_{i_l} has product equal to a square if and only if

$$\sum_{j=1}^l \underline{v}(x_{i_j}) = \underline{0}$$

- So, finding a product which is a square is equivalent to finding a linearly dependent set.



How to find a subset of the x_i for which the product of $f(x_{i_j})$ is a square?

- Start with a factor base \mathcal{B} of size k
- For each \mathcal{B} -smooth $f(x_i)$, create a binary exponent vector $\underline{v}(x_i) \in GF_2^k$ representing the parity of the powers of $p \in \mathcal{B}$ which divide $f(x_i)$.
- Then a subset x_{i_1}, \dots, x_{i_l} has product equal to a square if and only if

$$\sum_{j=1}^l \underline{v}(x_{i_j}) = \underline{0}$$

- So, finding a product which is a square is equivalent to finding a linearly dependent set.

The exponent vectors will always be regarded as row vectors





Trivial Linear Dependency

The following fact will be used over and over:

If we have a vector space of dimension k and if we have a set containing more than k vectors, then the set is linearly dependent.



Standard Method

- Take sufficiently many x_i that we obtain at least $k + 1 = |\mathcal{B}| + 1$ \mathcal{B} -smooth $f(x_i)$ values.
- Then we will certainly have a linearly dependent set
- Problem: if $|\mathcal{B}|$ is large, then the probability that a random $f(x_i)$ is \mathcal{B} -smooth is not too small, but we need many of them, and have to solve a very large linear system.
- Problem: if $|\mathcal{B}|$ is small, then we don't need to find many \mathcal{B} -smooth numbers, but the probability that a random $f(x_i)$ is \mathcal{B} -smooth is extremely small, so we need to sieve a very large number of values.



Standard Method

- Take sufficiently many x_i that we obtain at least $k + 1 = |\mathcal{B}| + 1$ \mathcal{B} -smooth $f(x_i)$ values.
- Then we will certainly have a linearly dependent set
- Problem: if $|\mathcal{B}|$ is large, then the probability that a random $f(x_i)$ is \mathcal{B} -smooth is not too small, but we need many of them, and have to solve a very large linear system.
- Problem: if $|\mathcal{B}|$ is small, then we don't need to find many \mathcal{B} -smooth numbers, but the probability that a random $f(x_i)$ is \mathcal{B} -smooth is extremely small, so we need to sieve a very large number of values.
- Perhaps we can do better: do we really need this many?



Do we need to factor this many $f(x_i)$ values?



14/59



Do we need to factor this many $f(x_i)$ values?

Pomerance asked: how many random vectors do we need to take in GF_2^k in order to obtain a linearly dependent set with reasonably high probability?



14/59



Do we need to factor this many $f(x_i)$ values?

Pomerance asked: how many random vectors do we need to take in GF_2^k in order to obtain a linearly dependent set with reasonably high probability?

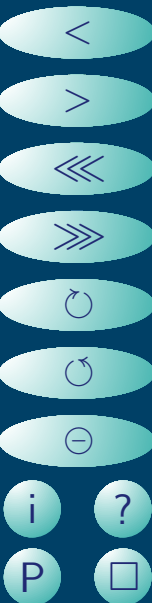
- Uniform distribution on all vectors in GF_2^k ?



Do we need to factor this many $f(x_i)$ values?

Pomerance asked: how many random vectors do we need to take in GF_2^k in order to obtain a linearly dependent set with reasonably high probability?

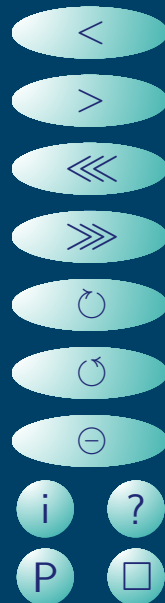
- Uniform distribution on all vectors in GF_2^k ?
- Uniform distribution on all vectors with h 1's?



Do we need to factor this many $f(x_i)$ values?

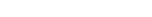
Pomerance asked: how many random vectors do we need to take in GF_2^k in order to obtain a linearly dependent set with reasonably high probability?

- Uniform distribution on all vectors in GF_2^k ?
- Uniform distribution on all vectors with h 1's?
- Other distributions?



The REU Problem

Motivation over: the problem now posed to the REU students became:





The REU Problem

Motivation over: the problem now posed to the REU students became:

Given a probability distribution on the set of vectors in GF_2^k , if we draw vectors (independently, with replacement, according to this distribution) at random, how many do we need to draw to have a high probability of a linear dependency?





The REU Problem

Motivation over: the problem now posed to the REU students became:

Given a probability distribution on the set of vectors in GF_2^k , if we draw vectors (independently, with replacement, according to this distribution) at random, how many do we need to draw to have a high probability of a linear dependency?

In particular, if we take a probability distribution which is similar to that of exponent vectors of numbers $x^2 - n$ where x is chosen uniformly from numbers not much larger than n , how many vectors do we need to obtain a linearly dependent set?





The REU Problem

Motivation over: the problem now posed to the REU students became:

Given a probability distribution on the set of vectors in GF_2^k , if we draw vectors (independently, with replacement, according to this distribution) at random, how many do we need to draw to have a high probability of a linear dependency?

In particular, if we take a probability distribution which is similar to that of exponent vectors of numbers $x^2 - n$ where x is chosen uniformly from numbers not much larger than n , how many vectors do we need to obtain a linearly dependent set?

The hope: that we may be able to get by with far fewer \mathcal{B} -smooth numbers in order to find a product which is a square.





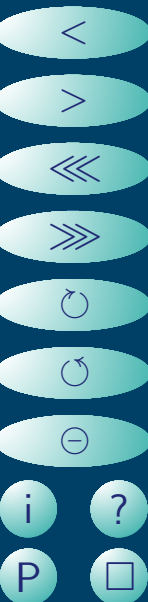
Uniform Distribution

Uniform distribution on all vectors in GF_2^k

- Easy to see: probability that k vectors are independent is

$$\prod_{j=1}^k (1 - 2^{-j})$$

- As $k \rightarrow \infty$ this converges to $0.288788\dots$
- This model doesn't improve the sieve at all





Uniform Distribution

Uniform distribution on all vectors in GF_2^k

- Easy to see: probability that k vectors are independent is

$$\prod_{j=1}^k (1 - 2^{-j})$$

- As $k \rightarrow \infty$ this converges to $0.288788\dots$
- This model doesn't improve the sieve at all
— but this model doesn't model reality!



Integers don't have many prime factors!

- The maximum number of distinct prime factors that an integer n can have is about $\log n / \log \log n$



17/59



Integers don't have many prime factors!

- The maximum number of distinct prime factors that an integer n can have is about $\log n / \log \log n$
- (Why? $\prod_{p < x} p \simeq \exp(x)$)



17/59



Integers don't have many prime factors!

- The maximum number of distinct prime factors that an integer n can have is about $\log n / \log \log n$
- (Why? $\prod_{p < x} p \simeq \exp(x)$)
- The typical number of prime factors that an integer n will have is about $\log \log n$. Hence if the number of primes in the factor base is large, the set of all vectors in GF_2^k is a bad model for the exponent vectors.





Vectors of Constant Weight: a Better Model?

- Typical integer near n has about $\log \log n$ prime factors.





Vectors of Constant Weight: a Better Model?

- Typical integer near n has about $\log \log n$ prime factors.
- If we choose vectors uniformly from the set of vectors with l 1's from GF_2^k , how many do we need to ensure that we have a dependent set?





Vectors of Constant Weight: a Better Model?

- Typical integer near n has about $\log \log n$ prime factors.
- If we choose vectors uniformly from the set of vectors with l 1's from GF_2^k , how many do we need to ensure that we have a dependent set?
- (With high probability)





Vectors of Constant Weight: a Better Model?

- Typical integer near n has about $\log \log n$ prime factors.
- If we choose vectors uniformly from the set of vectors with l 1's from GF_2^k , how many do we need to ensure that we have a dependent set?
- (With high probability)

Theorem: There are constants a and b so that if we have fewer than ak vectors, they are almost surely independent, and if we have more than bk vectors, they are almost surely dependent





Vectors of Constant Weight: a Better Model?

- Typical integer near n has about $\log \log n$ prime factors.
- If we choose vectors uniformly from the set of vectors with l 1's from GF_2^k , how many do we need to ensure that we have a dependent set?
- (With high probability)

Theorem: There are constants a and b so that if we have fewer than ak vectors, they are almost surely independent, and if we have more than bk vectors, they are almost surely dependent

(as $k \rightarrow \infty$)





Vectors of Constant Weight: a Better Model?

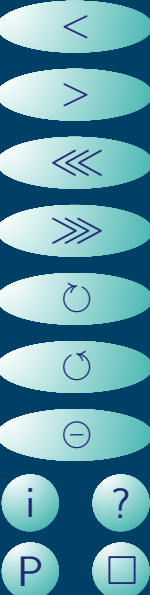
- Typical integer near n has about $\log \log n$ prime factors.
- If we choose vectors uniformly from the set of vectors with l 1's from GF_2^k , how many do we need to ensure that we have a dependent set?
- (With high probability)

Theorem: There are constants a and b so that if we have fewer than ak vectors, they are almost surely independent, and if we have more than bk vectors, they are almost surely dependent

(as $k \rightarrow \infty$)

- $a \simeq 1 - \frac{e^{-l}}{\log 2}$
- $b \simeq 1 - e^{-l}$

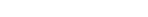
This model doesn't suggest that we can use significantly fewer \mathcal{B} -smooth numbers



Incidentally, this was essentially the jumping off point for the undergraduate students: to determine experimentally, if they could, which of the two bounds is actually the true threshold.

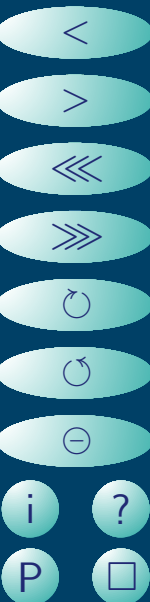


19/59



Incidentally, this was essentially the jumping off point for the undergraduate students: to determine experimentally, if they could, which of the two bounds is actually the true threshold.

Answer: Even with $k = 3$ the truth is hard to determine: it appears that the lower bound is the truth, but the data is not convincing.

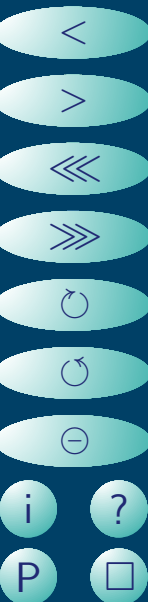




Incidentally, this was essentially the jumping off point for the undergraduate students: to determine experimentally, if they could, which of the two bounds is actually the true threshold.

Answer: Even with $k = 3$ the truth is hard to determine: it appears that the lower bound is the truth, but the data is not convincing.

Can't even prove that there *really is* a threshold function!



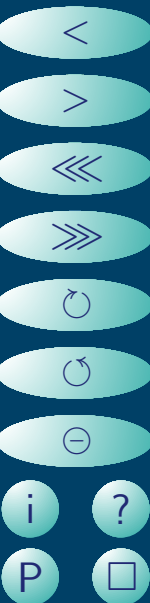


Incidentally, this was essentially the jumping off point for the undergraduate students: to determine experimentally, if they could, which of the two bounds is actually the true threshold.

Answer: Even with $k = 3$ the truth is hard to determine: it appears that the lower bound is the truth, but the data is not convincing.

Can't even prove that there *really is* a threshold function!

Many thanks to Jon Borwein for organizing time on a cluster at SFU



But this model is not a good model of reality either!



20/59



But this model is not a good model of reality either!

Why?



20/59



But this model is not a good model of reality either!

Why?

The “probability” a number is even is much greater than the “probability” it is divisible by, say, 97

So, vectors of constant weight have much lighter heads and much heavier tails than exponent vectors.



20/59





A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

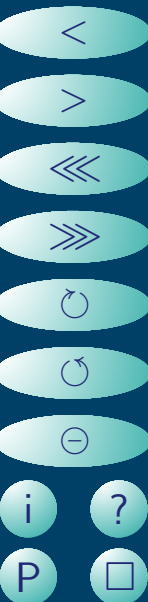




A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

$$\frac{1}{p}$$



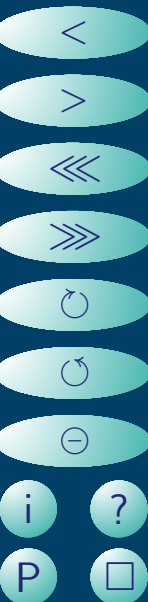


A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

$$\frac{1}{p}$$

- What is the probability that it is divisible by an odd power of p ?





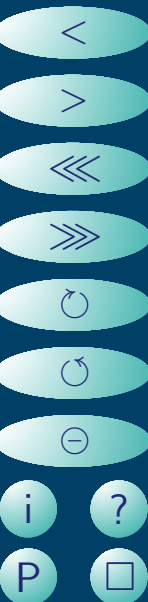
A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

$$\frac{1}{p}$$

- What is the probability that it is divisible by an odd power of p ?

$$\frac{1}{p+1}$$



A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

$$\frac{1}{p}$$

- What is the probability that it is divisible by an odd power of p ?

$$\frac{1}{p+1}$$

- Choose l vectors independently with $\underline{v}_1, \underline{v}_2, \underline{v}_3, \dots, \underline{v}_l \in GF_2^k$ as follows:

$$Pr(\underline{v}_i[j] = 1) = \frac{1}{p_j + 1}$$

where the factor base is $\mathcal{B} = \{p_1, p_2, \dots, p_k\}$

A Better Model

- If we take a number of size around n , what is the probability that it is divisible by p ?

$$\frac{1}{p}$$

- What is the probability that it is divisible by an odd power of p ?

$$\frac{1}{p+1}$$

- Choose l vectors independently with $\underline{v}_1, \underline{v}_2, \underline{v}_3, \dots, \underline{v}_l \in GF_2^k$ as follows:

$$Pr(\underline{v}_i[j] = 1) = \frac{1}{p_j + 1}$$

where the factor base is $\mathcal{B} = \{p_1, p_2, \dots, p_k\}$

Technical note: if $p|(x^2 - n)$ then n is a quadratic residue (mod p). This rules out about half of all small primes from being in \mathcal{B} , so

$$p_j \simeq 2j \log j.$$



In this model, how big should l be



23/59



In this model, how big should l be (as a function of k)



23/59



In this model, how big should l be (as a function of k) so that if we pick l vectors randomly in GF_2^k , then with high probability we get a linearly dependent set of vectors?





In this model, how big should l be (as a function of k) so that if we pick l vectors randomly in GF_2^k , then with high probability we get a linearly dependent set of vectors?

This question may be hard.





In this model, how big should l be (as a function of k) so that if we pick l vectors randomly in GF_2^k , then with high probability we get a linearly dependent set of vectors?

This question may be hard.

Can we get an upper bound?





In this model, how big should l be (as a function of k) so that if we pick l vectors randomly in GF_2^k , then with high probability we get a linearly dependent set of vectors?

This question may be hard.

Can we get an upper bound?

From now on, k and l will be assumed large, and we will definitely gloss over asymptotics.



Trivial Dependencies

Ways to show linear dependency:



24/59



Trivial Dependencies

Ways to show linear dependency:

Trivially dependent (TD)



24/59





Trivial Dependencies

Ways to show linear dependency:

Trivially dependent (TD)

Almost trivially dependent (ATD)





Trivial Dependencies

Ways to show linear dependency:

Trivially dependent (TD)

Almost trivially dependent (ATD)

Nearly trivially dependent (NTD)





Trivial Dependencies

Ways to show linear dependency:

Trivially dependent (TD)

Almost trivially dependent (ATD)

Nearly trivially dependent (NTD)

Somewhat trivially dependent (STD)





Trivial Dependencies

Ways to show linear dependency:

Trivially dependent (TD)

Almost trivially dependent (ATD)

Nearly trivially dependent (NTD)

Somewhat trivially dependent (STD)

Construct an array A from the row vectors $\underline{v}_1, \underline{v}_2, \dots, \underline{v}_l$





TD: trivially dependent

If A has more rows than columns, then the rows are *trivially dependent*. This is the standard method for ensuring a linear dependency in the exponent vectors for the Quadratic Sieve: find more than k \mathcal{B} -smooth numbers.





TD: trivially dependent

If A has more rows than columns, then the rows are *trivially dependent*. This is the standard method for ensuring a linear dependency in the exponent vectors for the Quadratic Sieve: find more than k \mathcal{B} -smooth numbers.

A trivially dependent 6×5 array

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$





ATD: almost trivially dependent

Suppose that some of the columns of A may be zero: then if we have more rows in A than we have non-zero columns, A is *almost trivially dependent*.



ATD: almost trivially dependent

Suppose that some of the columns of A may be zero: then if we have more rows in A than we have non-zero columns, A is *almost trivially dependent*.

A almost trivially dependent 6×7 array (6×5 after removal of [empty columns](#))

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$





NTD: nearly trivially dependent

If a column has exactly one 1 in it, we will refer to the 1 as a *solon*. Any row containing a solon can't be used in a linear dependency: hence if we remove all columns and rows which contain solons, and the remaining array has more rows than columns, then our new array is trivially dependent, and our original array A is *nearly trivially dependent*.

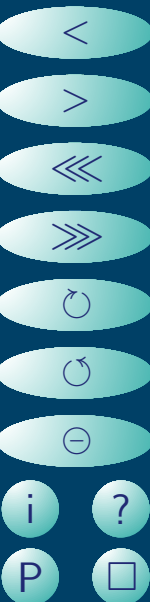




NTD: nearly trivially dependent

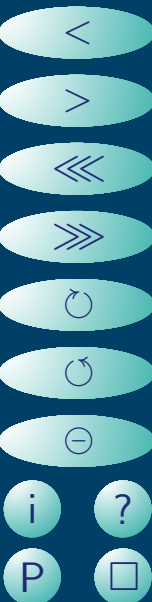
If a column has exactly one 1 in it, we will refer to the 1 as a *solon*. Any row containing a solon can't be used in a linear dependency: hence if we remove all columns and rows which contain solons, and the remaining array has more rows than columns, then our new array is trivially dependent, and our original array A is *nearly trivially dependent*.

Note: a column can only contain at most one solon: a row can contain many solons, so we will remove at least as many columns as rows.



An nearly trivially dependent 6×7 array (5×4 after removal of **columns** and **rows** containing **solons**)

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$





STD: somewhat trivial dependency

If a column has exactly two 1's (call the pair of 1's a *colon*), then the corresponding rows can only appear as part of a linear dependency together: hence we can replace the two rows of the array by their sum, and delete the column.

Again, columns contain at most one colon: however, rows can be involved in more than one colon. Model this with a graph $G = (V, E)$, where V is the set of rows, and the edges correspond to colons. Then the replacement of rows by their sum corresponds to replacing components in the graph by a single vertex. The original array is *somewhat trivially dependent* if there are more components (including isolated vertices) in the graph than there are columns remaining after deletion. If all components are trees, we gain nothing: however, every component containing a cycle means we lose at least one more column than row.



How can we use ATD's, NTD's and STD's to obtain upper bounds?



30/59



How many columns are empty?

The probability that the i th column of A is zero is

$$\left(1 - \frac{1}{p_i + 1}\right)^l \simeq \exp\left(-\frac{l}{2i \log i}\right)$$

Hence the expected number of non-zero columns is about

$$\sum_{i=0}^k \left(1 - \exp\left(-\frac{l}{2i \log i}\right)\right)$$

The number of non-zero columns will be fairly sharply concentrated about its mean, and hence if l is large enough that

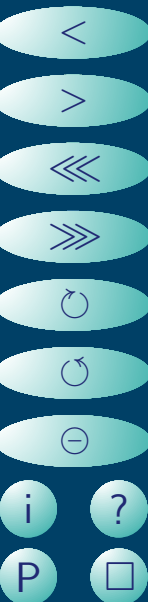
$$l > (1 + \epsilon) \sum_{i=0}^k \left(1 - \exp\left(-\frac{l}{2i \log i}\right)\right)$$

then with high probability the array will be almost trivially dependent.

When does this happen? Estimate the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i \log i} \right) \right)$$

by splitting it into three regions, and assuming that $l = k^\gamma$

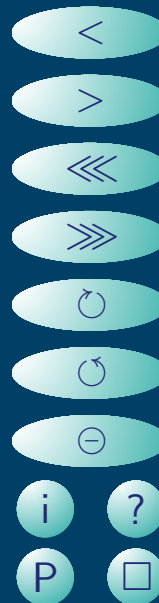


When does this happen? Estimate the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i \log i} \right) \right)$$

by splitting it into three regions, and assuming that $l = k^\gamma$

- (I) $1 \leq i \leq l/\log(l)^2$
- (II) $l/\log(l)^2 < i \leq l$
- (III) $l < i \leq k$



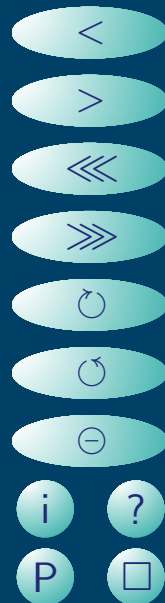
When does this happen? Estimate the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i \log i} \right) \right)$$

by splitting it into three regions, and assuming that $l = k^\gamma$

- (I) $1 \leq i \leq l/\log(l)^2$
- (II) $l/\log(l)^2 < i \leq l$
- (III) $l < i \leq k$

Easy to see that region (I) contributes $o(l)$ to the sum.



When does this happen? Estimate the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i \log i} \right) \right)$$

by splitting it into three regions, and assuming that $l = k^\gamma$

- (I) $1 \leq i \leq l/\log(l)^2$
- (II) $l/\log(l)^2 < i \leq l$
- (III) $l < i \leq k$

Easy to see that region (I) contributes $o(l)$ to the sum.

Estimate region (II) by approximating by an integral: it contributes $o(l)$ to the sum too.



When does this happen? Estimate the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i \log i} \right) \right)$$

by splitting it into three regions, and assuming that $l = k^\gamma$

- (I) $1 \leq i \leq l/\log(l)^2$
- (II) $l/\log(l)^2 < i \leq l$
- (III) $l < i \leq k$

Easy to see that region (I) contributes $o(l)$ to the sum.

Estimate region (II) by approximating by an integral: it contributes $o(l)$ to the sum too.

Estimate region (III) using Merten's theorem: it contributes about $-\log(\gamma)l$ to the sum: hence if $-\log(\gamma) < 1$, we get a high probability of an almost trivially dependent array. This gives a threshold of

$$l = k^{1/e}.$$





How good is this as a model of the quadratic sieve?



34/59





How good is this as a model of the quadratic sieve?

Send students away to code up the (relevant part of the) quadratic sieve, and get answer





How good is this as a model of the quadratic sieve?

Send students away to code up the (relevant part of the) quadratic sieve, and get answer

Not Very



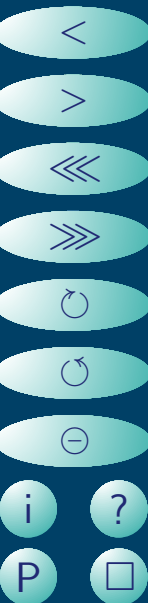


How good is this as a model of the quadratic sieve?

Send students away to code up the (relevant part of the) quadratic sieve, and get answer

Not Very

It appears that our model finds dependencies occurring far earlier than in reality.





How good is this as a model of the quadratic sieve?

Send students away to code up the (relevant part of the) quadratic sieve, and get answer

Not Very

It appears that our model finds dependencies occurring far earlier than in reality.

What is wrong? Our random arrays appear to be a little sparser than the arrays generated by the quadratic sieve: why is this?





How good is this as a model of the quadratic sieve?

Send students away to code up the (relevant part of the) quadratic sieve, and get answer

Not Very

It appears that our model finds dependencies occurring far earlier than in reality.

What is wrong? Our random arrays appear to be a little sparser than the arrays generated by the quadratic sieve: why is this?

What is the probability that a \mathcal{B} -smooth number is divisible by p ?





Adjusting the probabilities for smoothness

Does the fact that we are conditioning upon the fact that our $f(x_{i_j})$ values are \mathcal{B} -smooth have a significant effect on the probability that $p|f(x)$?





Adjusting the probabilities for smoothness

Does the fact that we are conditioning upon the fact that our $f(x_{i_j})$ values are \mathcal{B} -smooth have a significant effect on the probability that $p|f(x)$?

An extreme case: suppose that $\mathcal{B} = \{2\}$: what is the probability that a \mathcal{B} -smooth number y less than n is divisible by p ?





Adjusting the probabilities for smoothness

Does the fact that we are conditioning upon the fact that our $f(x_{i_j})$ values are \mathcal{B} -smooth have a significant effect on the probability that $p|f(x)$?

An extreme case: suppose that $\mathcal{B} = \{2\}$: what is the probability that a \mathcal{B} -smooth number y less than n is divisible by p ?

$$\Pr(p|y) = \begin{cases} 0 & \text{if } p > 2 \\ 1 - \frac{1}{\lceil \log_2 n + 1 \rceil} & \text{if } p = 2 \end{cases}$$

It is easy to compute (approximate) similar expressions for when \mathcal{B} is $\{2, 3\}$, $\{2, 3, 5\}$, etc.





Adjusting the probabilities for smoothness

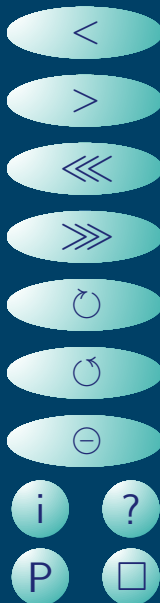
Does the fact that we are conditioning upon the fact that our $f(x_{i_j})$ values are \mathcal{B} -smooth have a significant effect on the probability that $p|f(x)$?

An extreme case: suppose that $\mathcal{B} = \{2\}$: what is the probability that a \mathcal{B} -smooth number y less than n is divisible by p ?

$$\Pr(p|y) = \begin{cases} 0 & \text{if } p > 2 \\ 1 - \frac{1}{\lfloor \log_2 n + 1 \rfloor} & \text{if } p = 2 \end{cases}$$

It is easy to compute (approximate) similar expressions for when \mathcal{B} is $\{2, 3\}$, $\{2, 3, 5\}$, etc.

Note the appearance of n as a parameter in the above expression.



Let

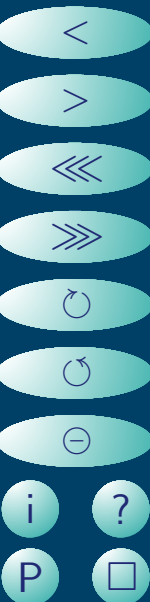
$$u = \frac{\log n}{\log k}$$

and

$$\delta = \frac{\log u}{\log k}$$

Then a reasonable estimate (in ranges of parameters we are likely to be interested in!), with $\mathcal{B} = \{p_1, p_2, \dots, p_k\}$, with \mathcal{B} containing about half the small primes, for $y < n$ and $p \in \mathcal{B}$, is

$$\Pr(p|y \text{ given that } y \text{ is } \mathcal{B}\text{-smooth}) \simeq \frac{1}{p^{1-\delta}}$$



Let

$$u = \frac{\log n}{\log k}$$

and

$$\delta = \frac{\log u}{\log k}$$

Then a reasonable estimate (in ranges of parameters we are likely to be interested in!), with $\mathcal{B} = \{p_1, p_2, \dots, p_k\}$, with \mathcal{B} containing about half the small primes, for $y < n$ and $p \in \mathcal{B}$, is

$$Pr(p|y \text{ given that } y \text{ is } \mathcal{B}\text{-smooth}) \simeq \frac{1}{p^{1-\delta}}$$

Incorporating this into the model is easy.





The model with smoothness incorporated

Choose l vectors independently with $\underline{v}_1, \underline{v}_2, \underline{v}_3, \dots, \underline{v}_l \in GF_2^k$ as follows:

$$Pr(\underline{v}_i[j] = 1) = \frac{1}{p_j^{1-\delta}}$$





The model with smoothness incorporated

Choose l vectors independently with $\underline{v}_1, \underline{v}_2, \underline{v}_3, \dots, \underline{v}_l \in GF_2^k$ as follows:

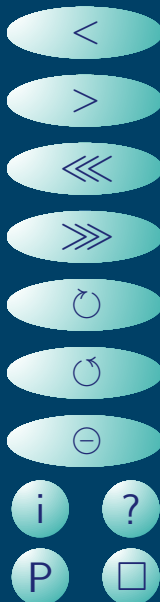
$$Pr(\underline{v}_i[j] = 1) = \frac{1}{p_j^{1-\delta}}$$

Again, note the dependence of this upon both n and k via the appearance of δ . In the regions that we are interested in, we typically have $\log k = \sqrt{\log n}$, so $u = \log k / \log n = \log k$, and

$$\delta = \frac{\log u}{\log k} = \frac{\log \log k}{\log k}.$$

Moreover most primes that will contribute to our estimates are of size similar to k : hence the extra contribution will be about

$$p^\delta \simeq k^{\frac{\log \log k}{\log k}} \simeq \log k.$$



If $i = ck$, then the i th prime in \mathcal{B} is about $2i \log i = 2ck(\log k + \log c)$, so for most primes in \mathcal{B} ,

$$\frac{1}{p_i^{1-\delta}} \simeq \frac{\log k}{2ck(\log k + \log c)} \simeq \frac{1}{2i}$$

This seemingly small change in the model is enough to change the qualitative behaviour: we no longer have the domination by the tail of the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i} \right) \right)$$

and in fact, we now need to consider nearly trivial dependencies (i.e. columns with exactly one 1) in order to obtain any improvement over the trivial bound.



If $i = ck$, then the i th prime in \mathcal{B} is about $2i \log i = 2ck(\log k + \log c)$, so for most primes in \mathcal{B} ,

$$\frac{1}{p_i^{1-\delta}} \simeq \frac{\log k}{2ck(\log k + \log c)} \simeq \frac{1}{2i}$$

This seemingly small change in the model is enough to change the qualitative behaviour: we no longer have the domination by the tail of the sum

$$\sum_{i=0}^k \left(1 - \exp \left(-\frac{l}{2i} \right) \right)$$

and in fact, we now need to consider nearly trivial dependencies (i.e. columns with exactly one 1) in order to obtain any improvement over the trivial bound.

Even more depressing: instead of obtaining $l = k^{1/e}$, we only get $l = Ck$ with $c \simeq 0.3$. While this *is* an improved bound, it is not of much practical help to the sieve.



But there is still hope!



39/59





But there is still hope!

We have not yet incorporated somewhat trivial dependencies (columns with exactly two 1's) into our analysis: this may help somewhat.

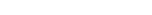




But there is still hope!

We have not yet incorporated somewhat trivial dependencies (columns with exactly two 1's) into our analysis: this may help somewhat.

Furthermore, when we delete rows and columns with solons, we can create new empty columns and new solons.





But there is still hope!

We have not yet incorporated somewhat trivial dependencies (columns with exactly two 1's) into our analysis: this may help somewhat.

Furthermore, when we delete rows and columns with solons, we can create new empty columns and new solons.

We need to model iterated deletions of solons and empty columns.

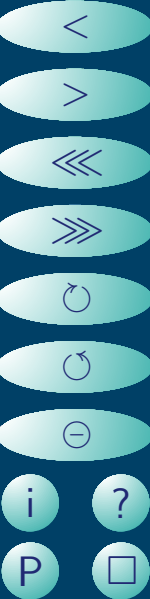


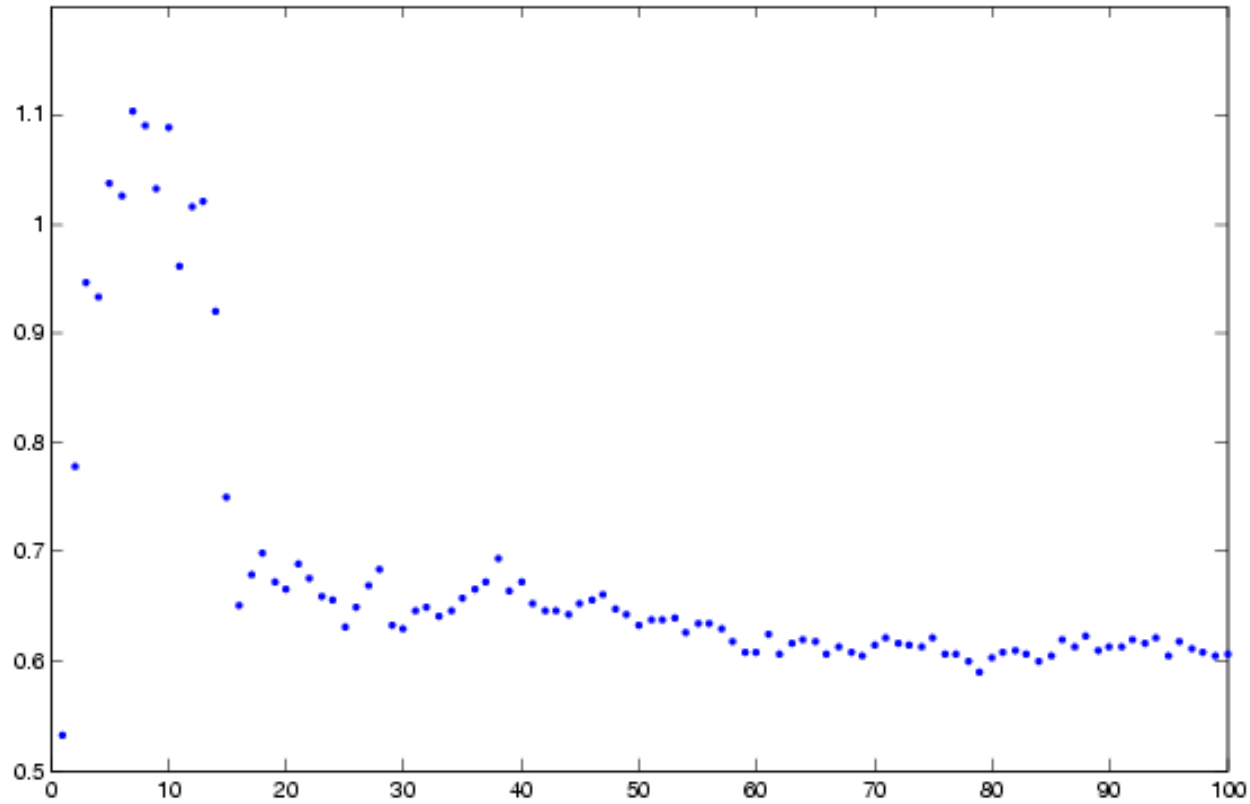


Empirical Data

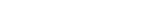
For a given n , compute l \mathcal{B} -smooth numbers and compute the proportion ρ_i which are divisible by each p_i . We expect this to behave like c/i .

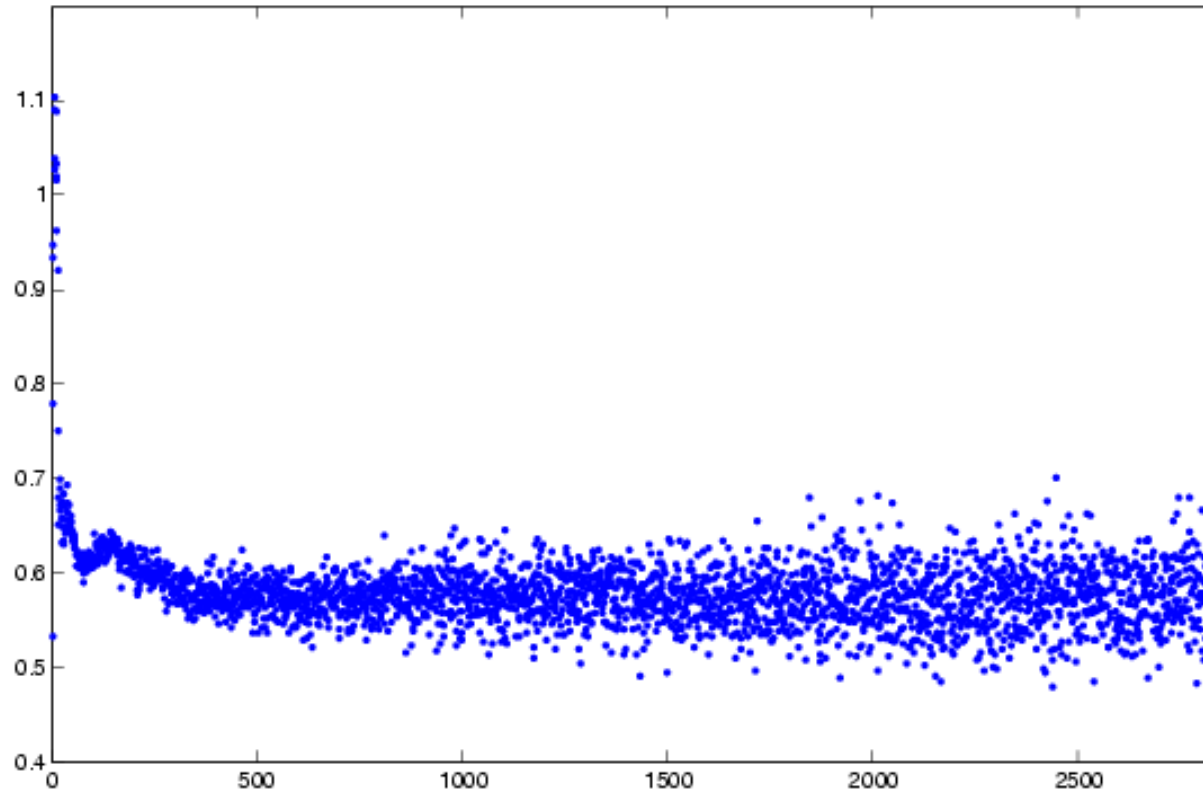
The following plots are of $i\rho_i$ for $n = 505167681297731$, and a factor base containing 2841 primes, when 513875 smooth values are computed.





Values of $i\rho_i$ for small primes

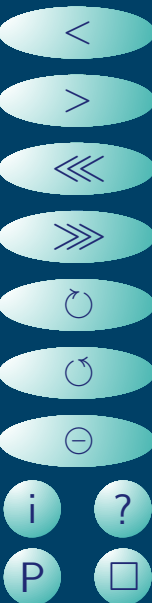


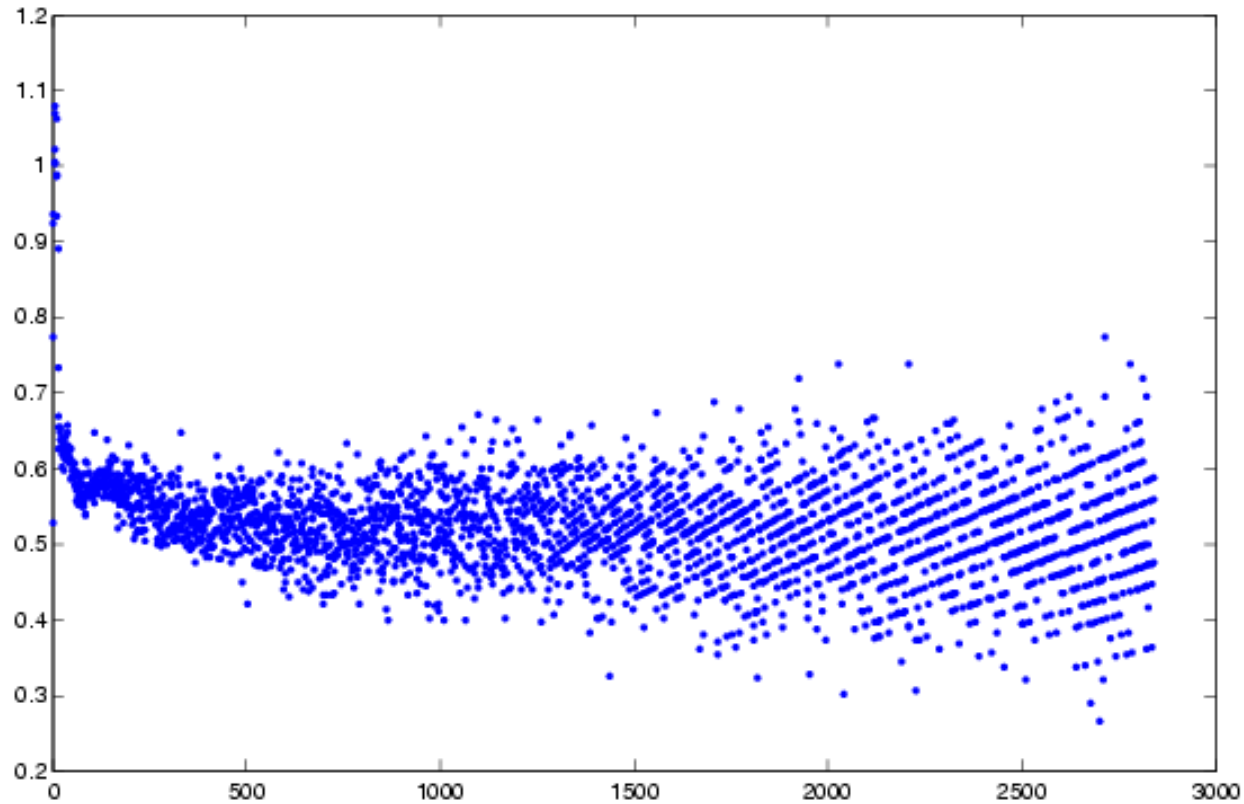


Values of $i\rho_i$ for large primes



Aside: interesting things happen when you don't have enough data: the following slide shows the values of $i\rho_i$ obtained for the same value of n but only 101679 smooth values





Values of $i\rho_i$ for large primes
too little data

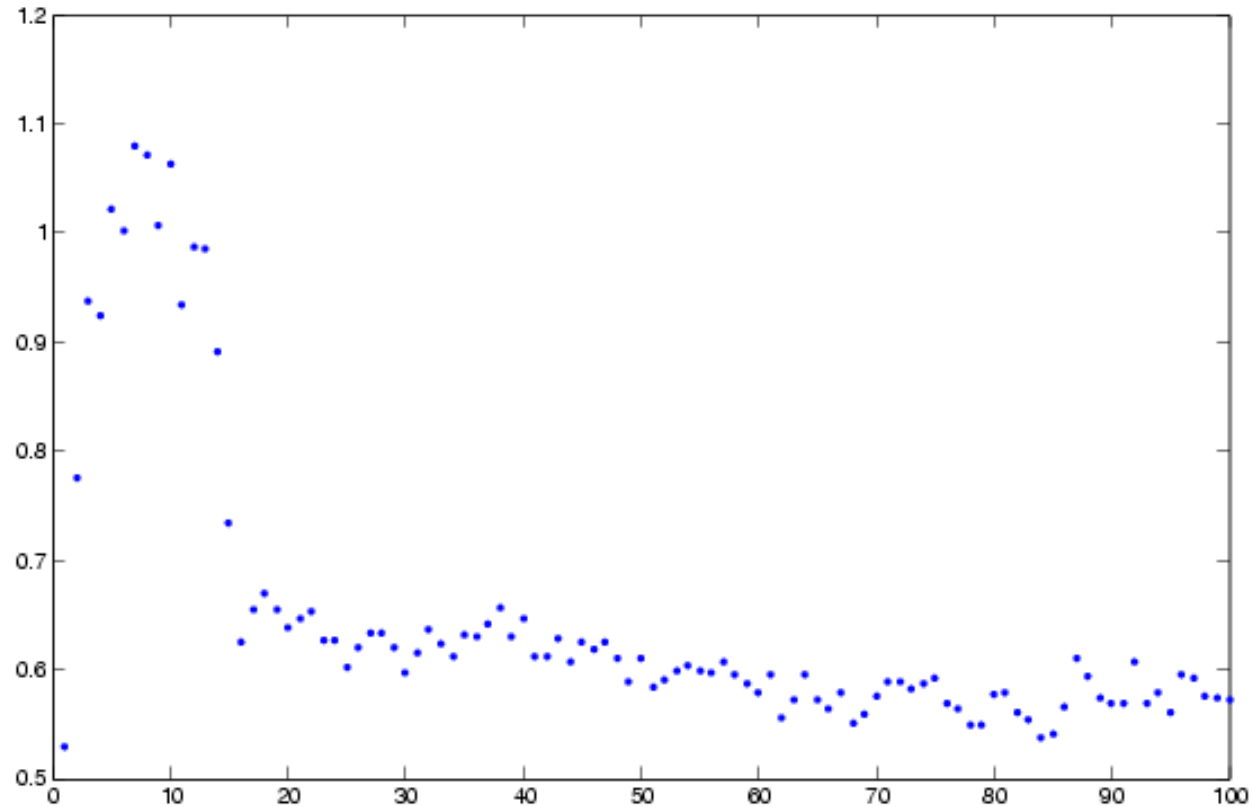


Note that the same effect is not seen if we just look at small primes!



45/59





Values of $i\rho_i$ for small primes
too little data



The data suggest that with the exception of small primes, the model

$$\alpha_i = \frac{c}{i}$$

is a good one, with c somewhat less than 0.7. Other slightly more simplistic estimates this summer by the REU students suggested that $c = 0.72$ was a reasonable model. Since we expect that increasing c decreases the probability of a linear dependency, in what follows we use the larger value.

We created arrays size $l \times k$ for various values of l, k , and removed colons, solons and empty columns until the remaining array had at least three ones in every column. The following results are typical





Comparing initial size and final size after solon and colon removal

Initial size	Final size
10000×10000	4700×2300
10000×15000	2250×1300
10000×20000	800×480
10000×25000	300×180
10000×30000	150×100
10000×35000	100×50
10000×40000	47×22
10000×45000	38×20





Comparing initial size and final size after solon and colon removal

Initial size	Final size
10000×10000	4700×2300
10000×15000	2250×1300
10000×20000	800×480
10000×25000	300×180
10000×30000	150×100
10000×35000	100×50
10000×40000	47×22
10000×45000	38×20

Notice that even though we are *increasing* the size of the initial array we are decreasing the size of the final array!





How large should l be as a function of k

We fixed $c = 0.72$ and constructed random arrays of size $l \times k$ for various values of l, k , removed solons and colons repeatedly, and computed various statistics:





How large should l be as a function of k

We fixed $c = 0.72$ and constructed random arrays of size $l \times k$ for various values of l, k , removed solons and colons repeatedly, and computed various statistics:

- the proportion of successful trials: trials for which iterated removal of solons and colons left an array with more non-zero rows than columns (implying the rows of the initial array are linearly dependent)





How large should l be as a function of k

We fixed $c = 0.72$ and constructed random arrays of size $l \times k$ for various values of l, k , removed solons and colons repeatedly, and computed various statistics:

- the proportion of successful trials: trials for which iterated removal of solons and colons left an array with more non-zero rows than columns (implying the rows of the initial array are linearly dependent)
- the proportion of trials for which iterated removal of solons and colons left a non-zero array with at most as many rows as columns (so that we can't conclude that the rows of the initial array are linearly dependent)





How large should l be as a function of k

We fixed $c = 0.72$ and constructed random arrays of size $l \times k$ for various values of l, k , removed solons and colons repeatedly, and computed various statistics:

- the proportion of successful trials: trials for which iterated removal of solons and colons left an array with more non-zero rows than columns (implying the rows of the initial array are linearly dependent)
- the proportion of trials for which iterated removal of solons and colons left a non-zero array with at most as many rows as columns (so that we can't conclude that the rows of the initial array are linearly dependent)
- the maximum and minimum number of non-zero rows and columns remaining after iterated removal of solons and colons





How large should l be as a function of k

We fixed $c = 0.72$ and constructed random arrays of size $l \times k$ for various values of l, k , removed solons and colons repeatedly, and computed various statistics:

- the proportion of successful trials: trials for which iterated removal of solons and colons left an array with more non-zero rows than columns (implying the rows of the initial array are linearly dependent)
- the proportion of trials for which iterated removal of solons and colons left a non-zero array with at most as many rows as columns (so that we can't conclude that the rows of the initial array are linearly dependent)
- the maximum and minimum number of non-zero rows and columns remaining after iterated removal of solons and colons
- the average number of rows and columns remaining after iterated removal of solons and colons.





We observed in almost all trials we either ended up with a zero matrix (so the original rows were independent) or with more rows than columns (so the original rows were dependent). This is consistent with the following conjecture:

Conjecture: Generate vectors in GF_2^k independently with $\alpha_j = 0.72/j$. Then almost surely (in the sense that the probability approaches 1 as $k \rightarrow \infty$) the first linear dependency gives a matrix which reduces to a $t+1 \times t$ matrix after iterated removal of solons and colons.

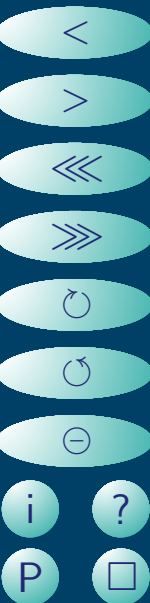
Essentially, this conjecture says that there are two conditions, one of which implies the other, and that usually, the two coincide exactly. This is a common situation in the theory of random graphs, in which for example, if the edges of the complete graph on n vertices are placed in random order, the edge which makes the graph connected is almost surely the edge which makes the minimum degree equal to 1.

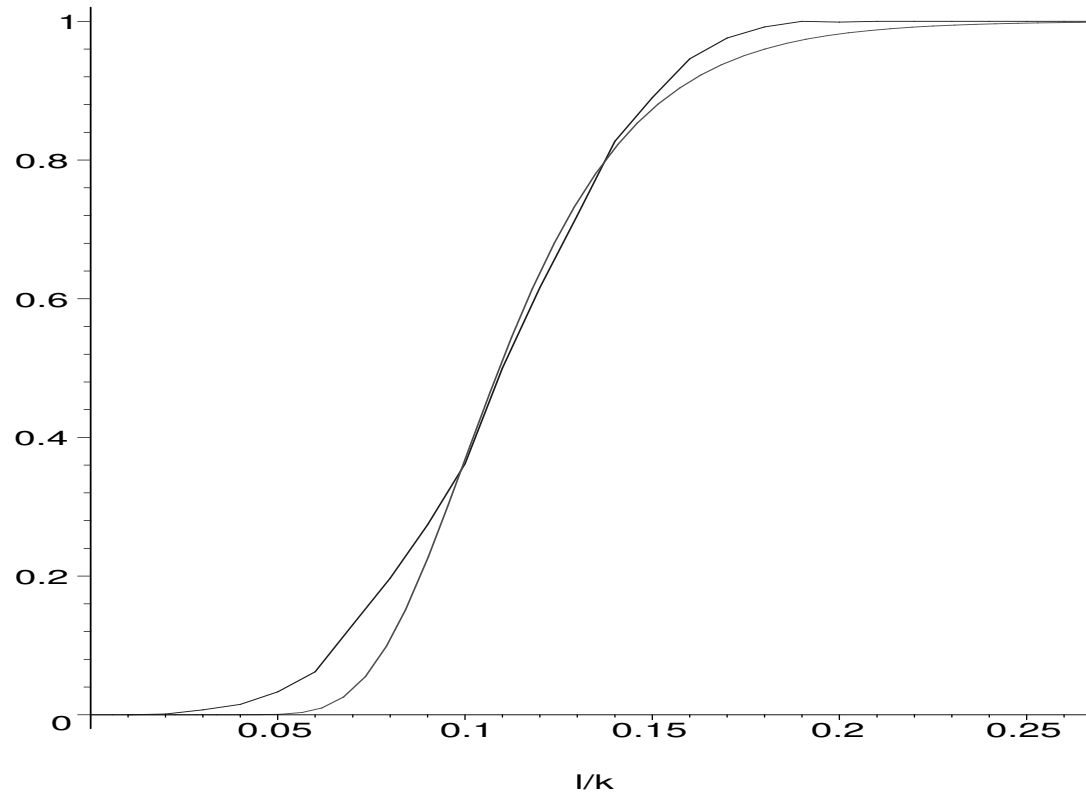




In studying threshold functions for the occurrence of combinatorial structures, for example in random graphs it is frequently the case that the threshold behaves (suitably scaled) like the function $e^{e^{-x}}$. The next figure shows the proportion of successful trials with $k = 20000$, for l from 1000, 1100, 1200, ... 3000, with 100 trials each, with an appropriately scaled and shifted copy of $e^{e^{-x}}$ overlaid.

We see that the match is close but not perfect. However, it is close enough to suggest that there is a similar threshold behaviour going on.

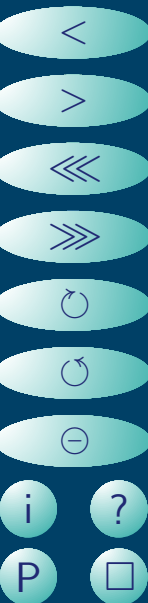




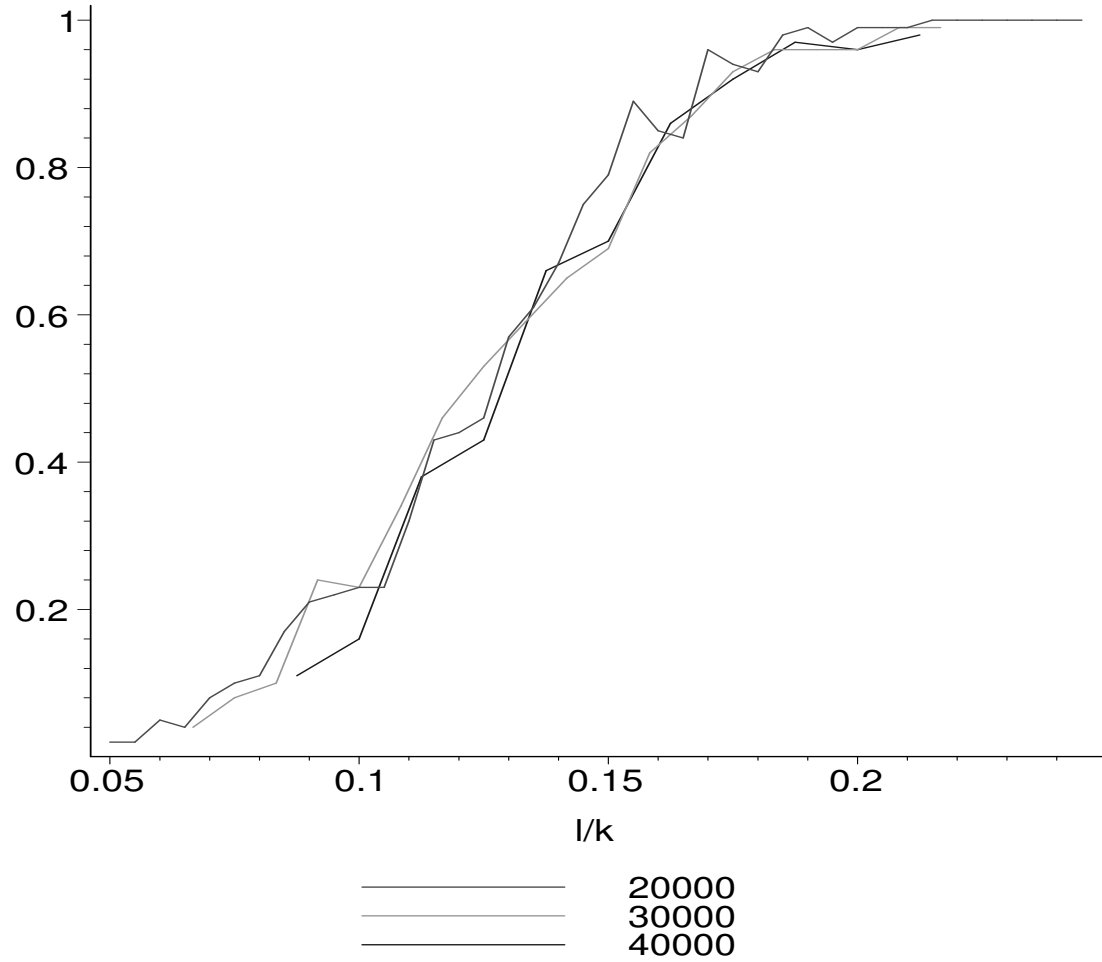
Transition for iterated deletion to show dependency



We created similar plots for $k = 20000, 30000$ and 40000 : the next figure shows them overlaid: it appears that the thresholds are similar in each case, although there may be a slight drift to the right as k increases.







Navigation controls:

- <
- >
- ⏪
- ⏩
- ↺
- ↻
- ⊖
- i
- ?
- P
-



Experimental evidence: factoring

Implementing the sieve on a small example: factoring a 14 digit number, with a factor base of $k = 2093$ primes, and $l = 429$ \mathcal{B} -smooth numbers, after iterated deletion of empty columns and solons, the resulting matrix has size 65×61 .

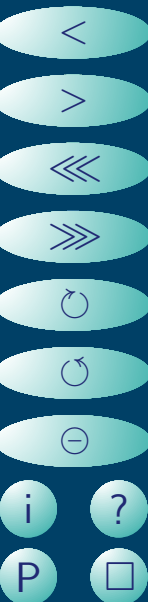




Experimental evidence: factoring

Implementing the sieve on a small example: factoring a 14 digit number, with a factor base of $k = 2093$ primes, and $l = 429$ \mathcal{B} -smooth numbers, after iterated deletion of empty columns and solons, the resulting matrix has size 65×61 .

Easier to solve than a 2094×2093 system!





Experimental evidence: factoring

Implementing the sieve on a small example: factoring a 14 digit number, with a factor base of $k = 2093$ primes, and $l = 429$ \mathcal{B} -smooth numbers, after iterated deletion of empty columns and solons, the resulting matrix has size 65×61 .

Easier to solve than a 2094×2093 system!

Not the smallest l which gives a dependency.





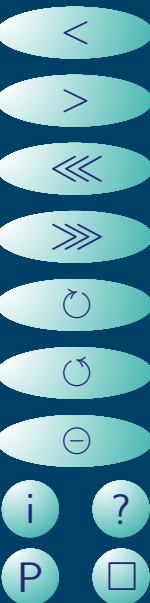
Experimental evidence: factoring

Implementing the sieve on a small example: factoring a 14 digit number, with a factor base of $k = 2093$ primes, and $l = 429$ \mathcal{B} -smooth numbers, after iterated deletion of empty columns and solons, the resulting matrix has size 65×61 .

Easier to solve than a 2094×2093 system!

Not the smallest l which gives a dependency.

Shows that trivial pre-processing of A really *does* reduce the size of the linear algebra problem to be tackled.



Conclusions

- Of the probabilistic models considered, it appears that a reasonable model for exponent vectors is

$$\alpha_i = \frac{c}{i}$$

where c is a constant between about 0.5 and 0.7.

- In this probabilistic model, if we have a $l \times k$ array, then (provably) we have dependence with high probability when $l > d(c)k$. The function d takes values, for example,

$$d(0.4) = 0.140$$

$$d(0.6) = 0.411$$

$$d(0.8) = 0.612$$



- In this probabilistic model, simulations suggest that linear dependency occurs earlier than this, and furthermore that by elimination of solons and colons, linear dependencies can be found very easily. If $l = d'(c)k$ then it appears (experimentally) that

$$d'(0.72) \simeq 0.2$$

and

$$d'(0.60) \simeq 0.1$$

- Initial investigations suggest that vectors produced by the Quadratic Sieve have the same behaviour, and that the sieve can be speeded up by taking this into account.
- A reason this has not apparently been noticed before is that if l is *large* as a function of k , then the elimination of solons and colons doesn't help very much at all! Standard implementations of the sieve have set $l > k$ to ensure linear dependence, and the effect will not be seen.



Still to be done



59/59



Still to be done

Apply these ideas to large factorization problems: do they really help?



59/59





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.

Analyze iterated deletion.





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.

Analyze iterated deletion.

Extend the model to handle somewhat trivial dependencies and iterated deletion.





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.

Analyze iterated deletion.

Extend the model to handle somewhat trivial dependencies and iterated deletion.

Analyze impact on the running time of the Quadratic Sieve with current parameters.





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.

Analyze iterated deletion.

Extend the model to handle somewhat trivial dependencies and iterated deletion.

Analyze impact on the running time of the Quadratic Sieve with current parameters.

Improve the parameter choices for the Quadratic Sieve.





Still to be done

Apply these ideas to large factorization problems: do they really help?

Prove or give good heuristic reasons why $\alpha_i = c/i$ is a good model.

Analyze iterated deletion.

Extend the model to handle somewhat trivial dependencies and iterated deletion.

Analyze impact on the running time of the Quadratic Sieve with current parameters.

Improve the parameter choices for the Quadratic Sieve.

Extend all this to the Number Field Sieve.

