An Early Idea on Factoring

Jevons Number

In the 1870s William Stanley Jevons wrote of the difficulty of factoring. We paraphrase Solomon Golomb's paraphrase:

> *Jevons observed that there are many cases where an operation is easy but its inverse is hard. He mentioned encryption and decryption. He mentioned multiplication and factoring. He anticipated RSA!*

Jevons Number

In the 1870s William Stanley Jevons wrote of the difficulty of factoring. We paraphrase Solomon Golomb's paraphrase:

> *Jevons observed that there are many cases where an operation is easy but its inverse is hard. He mentioned encryption and decryption. He mentioned multiplication and factoring. He anticipated RSA!*

Jevons thought factoring was hard (prob correct!) and that a certain number would never be factored (wrong!). Here is a quote:

Jevons Number

In the 1870s William Stanley Jevons wrote of the difficulty of factoring. We paraphrase Solomon Golomb's paraphrase:

> *Jevons observed that there are many cases where an operation is easy but its inverse is hard. He mentioned encryption and decryption. He mentioned multiplication and factoring. He anticipated RSA!*

Jevons thought factoring was hard (prob correct!) and that a certain number would never be factored (wrong!). Here is a quote:

> *Can the reader say what two numbers multiplied together will produce*
>
> $$8,616,460,799$$
>
> *I think it is unlikely that anyone aside from myself will ever know.*

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.
5. Golomb in 1996 showed that, given the math of his day, Jevons' number could be factored by hand.

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.
5. Golomb in 1996 showed that, given the math of his day, Jevons' number could be factored by hand.
6. Student: Why didn't Jevons just Google Factoring Quickly

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.
5. Golomb in 1996 showed that, given the math of his day, Jevons' number could be factored by hand.
6. Student: Why didn't Jevons just Google Factoring Quickly
   Bill: They didn't have the Web back then. Or Google.

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.
5. Golomb in 1996 showed that, given the math of his day, Jevons' number could be factored by hand.
6. Student: Why didn't Jevons just Google Factoring Quickly
   Bill: They didn't have the Web back then. Or Google.
   Student: How did they live?

Jevons Number

$$J = 8,616,460,799$$

We can now factor J easily. Was Jevons' comment stupid? Discuss

1. Jevons lived 1835–1882.
2. Jevons did not predict computers. Should he have?
3. Jevons did not predict math would help. Should he have?
4. Lehmer factored J in 1903 using math and computation.
5. Golomb in 1996 showed that, given the math of his day, Jevons' number could be factored by hand.
6. Student: Why didn't Jevons just Google Factoring Quickly
   Bill: They didn't have the Web back then. Or Google.
   Student: How did they live?
   Bill: How indeed!

Golomb's Method to Factor Jevons Number

$$J = 8,616,460,799$$

We apply a method of Fermat (in the 1600's) to the problem of factoring J.

To factor J find x, y such that

$$J = x^2 - y^2 = (x - y)(x + y)$$

So we must narrow our search for x, y.

Use Mods. Which Mod?

$$J = 8,616,460,799$$

Ends in 99. Hence

$$J \equiv 99 \equiv -1 \pmod{100}.$$

Use Mods. Which Mod?

$$J = 8,616,460,799$$

Ends in 99. Hence

$$J \equiv 99 \equiv -1 \pmod{100}.$$

Ah-ha. -1 is small! Mod 100 might be useful.

Golomb's Method to Factor Jevons Number

$$J = 8,616,460,799$$
$$J = x^2 - y^2$$

$$J \equiv x^2 - y^2 \pmod{100}$$

$$99 \equiv x^2 - y^2 \pmod{100}$$

$$y^2 + 99 \equiv x^2 \pmod{100}$$

$$y^2 \equiv x^2 - 99 \pmod{100}$$

$$y^2 \equiv x^2 + 1 \pmod{100}$$

Golomb's Method to Factor Jevons Number

$$J = 8,616,460,799$$
$$J = x^2 - y^2$$

$$J \equiv x^2 - y^2 \pmod{100}$$

$$99 \equiv x^2 - y^2 \pmod{100}$$

$$y^2 + 99 \equiv x^2 \pmod{100}$$

$$y^2 \equiv x^2 - 99 \pmod{100}$$

$$y^2 \equiv x^2 + 1 \pmod{100}$$

$$x^2 + 1 \equiv y^2 \pmod{100}$$

Golomb's Works Mod 100

$$x^2 + 1 \equiv y^2 \pmod{100}$$

All squares mod 100:

$$\{00, 01, 04, 09, 16, 21, 24, 25, 29, 36, 41, 44, 49\} \cup$$

$$\{56, 61, 64, 69, 76, 81, 84, 89, 96\}$$

The only pairs which differ by 1 are $(00, 01)$ and $(24, 25)$. So either:

1. $x^2 \equiv 0$, so $x \bmod 100 \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$
2. $x^2 \equiv 24$, so $x \bmod 100 \in \{18, 32, 68, 82\}$

WE SKIP NEXT FEW SLIDES

The next few slides are not hard, but they are tedious, so I keep them in this slide packet in case you want to look at them, but in class we'll skip them.

This material is NOT optional. It may be on a HW or Exam.

Golomb Works Mod 1000

$$x^2 - J \equiv y^2 \pmod{1000}$$

$$x^2 + 201 \equiv y^2 \pmod{1000}$$

If $x \pmod{100} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$ then
$x = 100a + 10b$
where $a \in \mathbb{N}$ and $b \in \{0, \ldots, 9\}$.
Easy but tedious to show that $b \equiv 0 \pmod 2$. Hence

1. $x^2 \equiv 0$, so $x \bmod 100 \in \{20, 40, 60, 80\}$
2. $x^2 \equiv 24$, so $x \bmod 100 \in \{18, 32, 68, 82\}$

Recap

Combine the two sets for $x \pmod{100}$ to get

$$x \pmod{100} \in \{18, 20, 32, 40, 60, 68, 80, 82\}$$

Since $J = x^2 - y^2$, $x^2 = J + y^2$, so

$$x \geq \left\lceil \sqrt{J} \right\rceil = 92824$$

Since $J = x^2 - y^2$, $x^2 - J = y^2$, hence

$$x^2 - J = y^2 \text{ a square}$$

Welcome BACK

After those tedious slides we have the next slide.

Golomb's Method to Factor Jevons Number: $x^2 \geq J$

1. $x \pmod{100} \in \{18, 20, 32, 40, 60, 68, 80, 82\}$
2. $x \geq \left\lceil \sqrt{J} \right\rceil = 92824$
3. $x^2 - J = y^2$, a square.

x	$y = (x^2 - J)^{1/2}$
92832	1148.6 ...
92840	1674.7 ...
92860	2553.1 ...
92868	2829.2 ...
92880	3199

AH-HA! We take $x = 92880$, $y = 3199$.

$$92880^2 - 3199^2 = 8,616,460,799$$

$$(92880 - 3199)(92880 + 3199) = 8,616,460,799$$

$$(89681)(96079) = 8,616,460,799$$

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.
3. Upshot He was in contact with math people and could have found a number theorist to ask. But he seems not to have.

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.
3. Upshot He was in contact with math people and could have found a number theorist to ask. But he seems not to have.

Did Jevons know about the work of Charles Babbage?

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.
3. Upshot He was in contact with math people and could have found a number theorist to ask. But he seems not to have.

Did Jevons know about the work of Charles Babbage?

1. Charles Babbage and Ada Lovelace were early computer scientists who worked together. (Calling them computer scientists is whiggish history.)

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.
3. Upshot He was in contact with math people and could have found a number theorist to ask. But he seems not to have.

Did Jevons know about the work of Charles Babbage?

1. Charles Babbage and Ada Lovelace were early computer scientists who worked together. (Calling them computer scientists is whiggish history.)
2. Charles Babbage also worked in Theology and wrote The Ninth Bridgewater Treatise. Jevons intended to write The Tenth Bridgewater Treatise.

What Math or CS Did Jevons Know or Know of?

Did Jevons ask any mathematicians about this?

1. Jevons worked in logic and knew De Morgan.
2. Jevons argued with Hermann von Helmholtz about non-Euclidean Geometry.
3. Upshot He was in contact with math people and could have found a number theorist to ask. But he seems not to have.

Did Jevons know about the work of Charles Babbage?

1. Charles Babbage and Ada Lovelace were early computer scientists who worked together. (Calling them computer scientists is whiggish history.)
2. Charles Babbage also worked in Theology and wrote The Ninth Bridgewater Treatise. Jevons intended to write The Tenth Bridgewater Treatise.
3. Upshot He knew who Babbage was and could have asked his opinion. But he seems not to have.

My Opinion and a Point

My Opinion and a Point

1. Jevons could have asked mathematicians about the Jevons Number, but didn't.
2. Jevons could have asked computer scientists (Babbage, Lovelace) about the Jevons Number, but didn't.
3. Jevons thought that since he couldn't have factored the Jevons Numbers if it was just given to him, nobody could.

My Opinion and a Point

1. Jevons could have asked mathematicians about the Jevons Number, but didn't.
2. Jevons could have asked computer scientists (Babbage, Lovelace) about the Jevons Number, but didn't.
3. Jevons thought that since he couldn't have factored the Jevons Numbers if it was just given to him, nobody could.

Many crypto systems are easily broken. Why? If Alice invents a crypto system that is easily broken then likely:

My Opinion and a Point

1. Jevons could have asked mathematicians about the Jevons Number, but didn't.
2. Jevons could have asked computer scientists (Babbage, Lovelace) about the Jevons Number, but didn't.
3. Jevons thought that since he couldn't have factored the Jevons Numbers if it was just given to him, nobody could.

Many crypto systems are easily broken. Why? If Alice invents a crypto system that is easily broken then likely:

1. Alice could have asked mathematicians about the Alice System, but didn't.
2. Alice could have asked computer scientists about the Alice System, but didn't.
3. Alice though that since she couldn't have broken Alice's system, nobody could.

My Opinion and a Point

1. Jevons could have asked mathematicians about the Jevons Number, but didn't.
2. Jevons could have asked computer scientists (Babbage, Lovelace) about the Jevons Number, but didn't.
3. Jevons thought that since he couldn't have factored the Jevons Numbers if it was just given to him, nobody could.

Many crypto systems are easily broken. Why? If Alice invents a crypto system that is easily broken then likely:

1. Alice could have asked mathematicians about the Alice System, but didn't.
2. Alice could have asked computer scientists about the Alice System, but didn't.
3. Alice though that since she couldn't have broken Alice's system, nobody could.

A lesson for us all!

Erik's Opinion

Erik, one of the TA's, when proofreading these slides, said the following:

Erik's Opinion

Erik, one of the TA's, when proofreading these slides, said the following:

Erik's Opinion

Erik, one of the TA's, when proofreading these slides, said the following:

1. Reasonable that he didn't realize that computers would get so much better.

Erik's Opinion

Erik, one of the TA's, when proofreading these slides, said the following:

1. Reasonable that he didn't realize that computers would get so much better.
2. Foolish since $J = 8,616,460,799$ isn't THAT big. Someone with enough determination could divide J by $2, 3, \ldots, \lceil \sqrt{J} \rceil$. This is only $\lceil \sqrt{J} \rceil = 92825$ trial divisions. Leave it to you to see if this is reasonable to finish in (say) 1 year.

My Opinion and a Counterpoint

Conjecture Jevons was arrogant. Likely true.

My Opinion and a Counterpoint

Conjecture Jevons was arrogant. Likely true.

Conjecture We have the arrogance of hindsight.

My Opinion and a Counterpoint

Conjecture Jevons was arrogant. Likely true.
Conjecture We have the arrogance of hindsight.

- It's easy for us to say
  What a moron! He should have asked a Number Theorist
  What was he going to do, Google Number Theorist ?

My Opinion and a Counterpoint

Conjecture Jevons was arrogant. Likely true.
Conjecture We have the arrogance of hindsight.

- It's easy for us to say
  What a moron! He should have asked a Number Theorist
  What was he going to do, Google Number Theorist ?
- It's easy for us to say
  What a moron! He should have asked a Babbage or Lovelace
  We know about the role of computers to speed up calculations, but it's reasonable it never dawned on him.

My Opinion and a Counterpoint

Conjecture Jevons was arrogant. Likely true.
Conjecture We have the arrogance of hindsight.

- It's easy for us to say
  What a moron! He should have asked a Number Theorist
  What was he going to do, Google Number Theorist ?
- It's easy for us to say
  What a moron! He should have asked a Babbage or Lovelace
  We know about the role of computers to speed up calculations, but it's reasonable it never dawned on him.
- Conclusion
  - His arrogance: assumed the world would not change much.
  - Our arrogance: knowing how much the world did change.

Factoring Algorithms

Recall Factoring Algorithm Ground Rules

- We only consider algorithms that, given N, find a non-trivial factor of N.
- We measure the run time as a function of $\lg N$ which is the *length* of the input. We may use L for this.
- We count $+$, $-$, $\times$, $\div$ as ONE step. A more refined analysis would count them as $(\lg x)^2$ steps where x is the largest number you are dealing with.
- We leave out the O-of but always mean O-of
- We leave out the *expected time* but always mean it. Our algorithms are randomized.

Recall Easy Factoring Algorithm

1. Input(N)
2. For $x = 2$ to $\lfloor N^{1/2} \rfloor$
   If x divides N then return x (and jump out of loop!).

This takes time $N^{1/2} = 2^{L/2}$.

Recall Easy Factoring Algorithm

1. Input(N)
2. For $x = 2$ to $\lfloor N^{1/2} \rfloor$

   If x divides N then return x (and jump out of loop!).

This takes time $N^{1/2} = 2^{L/2}$.

Goal Do much better than time $N^{1/2}$.
How Much Better? Ignoring (1) constants, (2) the lack of proofs of the runtimes, and (3) cheating a byte, we have:

- Easy: $N^{1/2} = 2^{L/2}$.
- Today's lecture: $N^{1/4} = 2^{L/4}$.
- Tomorrow's lecture: $N^{1/L^{1/2}} = 2^{L^{1/2}}$.
- Best Known: $N^{1/L^{2/3}} = 2^{L^{1/3}}$.

Pollard's ρ Algorithm for Factoring (1975)

Thought Experiment

We want to factor N.

Thought Experiment

We want to factor N.

p is smallest factor of N (we don't know p). Note $p \leq N^{1/2}$.

Thought Experiment

We want to factor N.

p is smallest factor of N (we don't know p). Note $p \leq N^{1/2}$.

We somehow find x, y such that $x \equiv y \pmod{p}$. Useful?

Thought Experiment

We want to factor N.

p is smallest factor of N (we don't know p). Note $p \leq N^{1/2}$.

We somehow find x, y such that $x \equiv y \pmod{p}$. Useful?

$\gcd(x - y, N)$ will likely yield a nontrivial factor of N since p divides both.

Thought Experiment

We want to factor N.

p is smallest factor of N (we don't know p). Note $p \leq N^{1/2}$.

We somehow find x, y such that $x \equiv y \pmod{p}$. Useful?

$\gcd(x - y, N)$ will likely yield a nontrivial factor of N since p divides both.

We look at several approaches to finding such an x, y that do not work before presenting the approach that does work.

Approach 1: Rand Seq mod p, Intuition

Generate random sequence $x_1, x_2, \ldots \in \{0, \ldots, N-1\}$.

Every time you get a new x_i, test, for all $1 \le j \le i-1$,

$$x_i \equiv x_j \pmod{p}.$$

Hope to get a YES.

If get YES then do

$$\gcd(x_i - x_j, N).$$

Approach One: Rand Seq mod p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while TRUE
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
        for $j \leftarrow 1$ to $i-1$
            if $x_i \equiv x_j \pmod{p}$ then
                $d \leftarrow \mathrm{gcd}(x_i - x_j, N)$
                if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i+1$
output(d)

Approach One: Rand Seq mod p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while TRUE
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
        for $j \leftarrow 1$ to $i-1$
            if $x_i \equiv x_j \pmod{p}$ then
                $d \leftarrow \gcd(x_i - x_j, N)$
                if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i + 1$
output(d)

PRO: Bday paradox: x_i's are balls, mod p are boxes. So likely to find $x_i \equiv x_j \pmod{p}$ within $p^{1/2} \sim N^{1/4}$ iterations.

Approach One: Rand Seq mod p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while TRUE
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
        for $j \leftarrow 1$ to $i-1$
            if $x_i \equiv x_j \pmod{p}$ then
                $d \leftarrow \gcd(x_i - x_j, N)$
                if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i+1$
output(d)

PRO: Bday paradox: x_i's are balls, mod p are boxes. So likely to find $x_i \equiv x_j \pmod{p}$ within $p^{1/2} \sim N^{1/4}$ iterations.

CON: Need to already know p. Really! Darn!

Approach One: Rand Seq mod p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while TRUE
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
        for $j \leftarrow 1$ to $i-1$
            if $x_i \equiv x_j \pmod{p}$ then
                $d \leftarrow \gcd(x_i - x_j, N)$
                if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i+1$
output(d)

PRO: Bday paradox: x_i's are balls, mod p are boxes. So likely to find $x_i \equiv x_j \pmod{p}$ within $p^{1/2} \sim N^{1/4}$ iterations.

CON: Need to already know p. Really! Darn!

ADJUST: Always do GCD.

Approach 2: Rand Seq mod p, W/O p, Intuition

Generate random sequence $x_1, x_2, \ldots \in \{0, \ldots, N-1\}$.

Every time you get a new x_i, do, for all $1 \le j \le i-1$,

$$\gcd(x_i - x_j, N).$$

So do not need to know p. And if $x_i \equiv x_j \pmod{p}$, you'll get a factor.

Approach 2: Rand Seq mod p, W/O p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$ $i \leftarrow 2$
while true
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
    for $j \leftarrow 1$ to $i-1$
        $d = \mathrm{gcd}(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
        $i \leftarrow i+1$
output(d)

Approach 2: Rand Seq mod p, W/O p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$ $i \leftarrow 2$
while true
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
    for $j \leftarrow 1$ to $i-1$
        $d = \gcd(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
        $i \leftarrow i+1$
output(d)

PRO: Bday paradox: x_i's:balls, mod p:boxes. Prob find $x_i \equiv x_j \pmod{p}$ with $i \leq p^{1/2} \sim N^{1/4}$. Perhaps sooner–other prime factors. Not knowing p does not matter.

Approach 2: Rand Seq mod p, W/O p, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$ $i \leftarrow 2$
while true
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
    for $j \leftarrow 1$ to $i-1$
        $d = \mathrm{gcd}(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
        $i \leftarrow i+1$
output(d)

PRO: Bday paradox: x_i's:balls, mod p:boxes. Prob find $x_i \equiv x_j \pmod{p}$ with $i \le p^{1/2} \sim N^{1/4}$. Perhaps sooner–other prime factors. Not knowing p does not matter.

CON: Iteration i makes i^2 operations. Total number of operations:

$$\sum_{i=1}^{N^{1/4}} i^2 \sim (N^{1/4})^3 \sim N^{3/4} \text{ BAD :-(.}$$

Another Issue: Space

$x_1 \leftarrow \mathrm{rand}(0, N-1)$ $i \leftarrow 2$
while true
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
    for $j \leftarrow 1$ to $i-1$
        $d = \gcd(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
        $i \leftarrow i + 1$
output(d)

Another Issue: Space

$x_1 \leftarrow \mathrm{rand}(0, N-1)$ $i \leftarrow 2$
while true
    $x_i \leftarrow \mathrm{rand}(0, N-1)$
    for $j \leftarrow 1$ to $i-1$
        $d = \gcd(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
        $i \leftarrow i+1$
output(d)

CON: After Iteration i need to store $x_1, \ldots, x_i$. Since $\sim N^{1/4}$ iterations this is $N^{1/4}$ space. Too much space :-(

Approach 3: Rand Looking Sequence, Intuition

How to create a random looking sequence?

Approach 3: Rand Looking Sequence, Intuition

How to create a random looking sequence?

- Pick random $x_1, c \in \{1, \ldots, N-1\}$

Approach 3: Rand Looking Sequence, Intuition

How to create a random looking sequence?

- Pick random $x_1, c \in \{1, \ldots, N-1\}$
- If know x_{i-1}, create

$$x_i = x_{i-1} * x_{i-1} + c \pmod{N}.$$

- The sequence x_1, x_2, x_3 will hopefully be random enough that the bday paradox applies. We use the informal term random looking for this.

Approach 3: Rand Looking Sequence, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $c \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while true
    $x_i \leftarrow x_{i-1} * x_{i-1} + c \pmod N$
    for $j \leftarrow 1$ to $i-1$
        for $k \leftarrow 2$ to j $x_k \leftarrow x_{k-1} * x_{k-1} + c$
        $d \leftarrow \gcd(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i + 1$
output(d)

Approach 3: Rand Looking Sequence, Program

$x_1 \leftarrow \mathrm{rand}(0, N-1)$, $c \leftarrow \mathrm{rand}(0, N-1)$, $i \leftarrow 2$
while true
    $x_i \leftarrow x_{i-1} * x_{i-1} + c \pmod{N}$
    for $j \leftarrow 1$ to $i-1$
        for $k \leftarrow 2$ to j $x_k \leftarrow x_{k-1} * x_{k-1} + c$
        $d \leftarrow \mathrm{gcd}(x_i - x_j, N)$
        if $d \neq 1$ and $d \neq N$ then break
    $i \leftarrow i+1$
output(d)

PRO Empirically seq x_1, x_2 is random enough, so $N^{1/4}$ iterations.
PRO Space not a problem.
CON Time still a problem :-(

What Do we Really Want?

Let $y_i \equiv x_i \pmod{p}$. $y_1, y_2, \ldots$ is random looking.
we want to find $i, j \leq N^{1/4}$ such that $y_i \equiv y_j \pmod{p}$.

What Do we Really Want?

Let $y_i \equiv x_i \pmod{p}$. $y_1, y_2, \ldots$ is random looking.
we want to find $i, j \leq N^{1/4}$ such that $y_i \equiv y_j \pmod{p}$.
Key y_i computed via recurrence so $y_i = y_j \implies y_{i+a} = y_{j+a}$.

What Do we Really Want?

Let $y_i \equiv x_i \pmod{p}$. $y_1, y_2, \ldots$ is random looking.
we want to find $i, j \leq N^{1/4}$ such that $y_i \equiv y_j \pmod{p}$.
Key y_i computed via recurrence so $y_i = y_j \implies y_{i+a} = y_{j+a}$.

Lemma If exists $i < j \leq M$ with $y_i = y_j$ then exists $k \leq 2M$ such that $y_k = y_{2k}$.
Proof Sketch $1 \leq i < j \leq M$ and $y_i = y_j$. For all a, $y_{i+a} = y_{j+a}$.

What Do we Really Want?

Let $y_i \equiv x_i \pmod{p}$. $y_1, y_2, \ldots$ is random looking.
we want to find $i, j \le N^{1/4}$ such that $y_i \equiv y_j \pmod{p}$.
Key y_i computed via recurrence so $y_i = y_j \implies y_{i+a} = y_{j+a}$.

Lemma If exists $i < j \le M$ with $y_i = y_j$ then exists $k \le 2M$ such that $y_k = y_{2k}$.
Proof Sketch $1 \le i < j \le M$ and $y_i = y_j$. For all a, $y_{i+a} = y_{j+a}$.

We need an a such that $j + a = 2(i + a)$. $a = j - 2i$ works.
$y_{i+(j-2i)} = y_{j+(j-2i)}$ hence $y_{j-i} = y_{2(j-i)}$. And $j - i \le M$.
Looks good! Is this proof correct?

What Do we Really Want?

Let $y_i \equiv x_i \pmod{p}$. $y_1, y_2, \ldots$ is random looking.
we want to find $i, j \le N^{1/4}$ such that $y_i \equiv y_j \pmod{p}$.
Key y_i computed via recurrence so $y_i = y_j \implies y_{i+a} = y_{j+a}$.

Lemma If exists $i < j \le M$ with $y_i = y_j$ then exists $k \le 2M$ such that $y_k = y_{2k}$.
Proof Sketch $1 \le i < j \le M$ and $y_i = y_j$. For all a, $y_{i+a} = y_{j+a}$.

We need an a such that $j + a = 2(i + a)$. $a = j - 2i$ works.
$y_{i+(j-2i)} = y_{j+(j-2i)}$ hence $y_{j-i} = y_{2(j-i)}$. And $j - i \le M$.
Looks good! Is this proof correct?

No What if $j - 2i \le 0$? Then does not work. Leave you to work out the details of that case.
End of Proof

Recap

Rand Looking Sequence x_1, c chosen at random in $\{1, \ldots, N\}$, then $x_i = x_{i-1} * x_{i-1} + c \pmod{N}$.

Recap

Rand Looking Sequence x_1, c chosen at random in $\{1, \ldots, N\}$, then $x_i = x_{i-1} * x_{i-1} + c \pmod{N}$.

We want to find i, j such $x_i \equiv x_j \pmod{p}$.

Recap

Rand Looking Sequence x_1, c chosen at random in $\{1, \ldots, N\}$, then $x_i = x_{i-1} * x_{i-1} + c \pmod{N}$.

We want to find i, j such $x_i \equiv x_j \pmod{p}$.

Don't know p. Really want $\gcd(x_i - x_j, N) \neq 1$.

Recap

Rand Looking Sequence x_1, c chosen at random in $\{1, \ldots, N\}$, then $x_i = x_{i-1} * x_{i-1} + c \pmod N$.

We want to find i, j such $x_i \equiv x_j \pmod p$.

Don't know p. Really want $\gcd(x_i - x_j, N) \neq 1$.

Trying all pairs is too much time.
Important If there is a pair then there is a pair of form x_i, x_{2i}.

Recap

Rand Looking Sequence x_1, c chosen at random in $\{1, \ldots, N\}$, then $x_i = x_{i-1} * x_{i-1} + c \pmod{N}$.

We want to find i, j such $x_i \equiv x_j \pmod{p}$.

Don't know p. Really want $\gcd(x_i - x_j, N) \neq 1$.

Trying all pairs is too much time.
Important If there is a pair then there is a pair of form x_i, x_{2i}.

Idea Only try pairs of form (x_i, x_{2i}).

Final Algorithm

Define $f_c(x) \leftarrow x * x + c$

$x \leftarrow \mathrm{rand}(0, N-1)$, $c \leftarrow \mathrm{rand}(0, N-1)$, $y \leftarrow f_c(x)$
while TRUE
    $x \leftarrow f_c(x)$
    $y \leftarrow f_c(f_c(y))$
    $d \leftarrow \gcd(x - y, N)$
    if $d \neq 1$ and $d \neq N$ then break
output(d)

Final Algorithm

Define $f_c(x) \leftarrow x * x + c$

$x \leftarrow \mathrm{rand}(0, N-1)$, $c \leftarrow \mathrm{rand}(0, N-1)$, $y \leftarrow f_c(x)$
while TRUE
    $x \leftarrow f_c(x)$
    $y \leftarrow f_c(f_c(y))$
    $d \leftarrow \gcd(x-y, N)$
    if $d \neq 1$ and $d \neq N$ then break
output(d)
PRO By Bday Paradox will likely finish in $N^{1/4}$ steps.
CON No real cons, but is $N^{1/4}$ fast enough?

How Good In Practice?

- The Algorithm is GOOD. Variations are GREAT.
- Was used to provide first factorization of $2^{2^8} + 1$.
- In 1975 was fastest algorithm in practice. Not anymore.
- Called *Pollard's ρ Algorithm* since he set $\rho = j - i$.
- Why we think $N^{1/4}$: Sequence seems random enough for Bday paradox to work.
- Why still unproven:

How Good In Practice?

- The Algorithm is GOOD. Variations are GREAT.
- Was used to provide first factorization of $2^{2^8}+1$.
- In 1975 was fastest algorithm in practice. Not anymore.
- Called *Pollard's ρ Algorithm* since he set $\rho = j - i$.
- Why we think $N^{1/4}$: Sequence seems random enough for Bday paradox to work.
- Why still unproven:
  - Proving that a deterministic sequence is random enough is hard to do or even define.

How Good In Practice?

- The Algorithm is GOOD. Variations are GREAT.
- Was used to provide first factorization of $2^{2^8}+1$.
- In 1975 was fastest algorithm in practice. Not anymore.
- Called *Pollard's ρ Algorithm* since he set $\rho = j - i$.
- Why we think $N^{1/4}$: Sequence seems random enough for Bday paradox to work.
- Why still unproven:
  - Proving that a deterministic sequence is random enough is hard to do or even define.
  - Natalie, Natalie, and Maddy haven't worked on it yet.

The Old Saying in Reverse

Typically one hears the following about academic research:

It works in theory, can we make it work in practice?

The Old Saying in Reverse

Typically one hears the following about academic research:

It works in theory, can we make it work in practice?

Pollard's ρ-algorithm is an example of the converse:

It works in practice, can we make it work in theory?

The Old Saying in Reverse

Typically one hears the following about academic research:

It works in theory, can we make it work in practice?

Pollard's ρ-algorithm is an example of the converse:

It works in practice, can we make it work in theory?

Why is it important to learn why it works in theory?

The Old Saying in Reverse

Typically one hears the following about academic research:

It works in theory, can we make it work in practice?

Pollard's ρ-algorithm is an example of the converse:

It works in practice, can we make it work in theory?

Why is it important to learn why it works in theory?

1. Make sure it really works. This is low-priority. Hey! It works!

The Old Saying in Reverse

Typically one hears the following about academic research:

It works in theory, can we make it work in practice?

Pollard's ρ-algorithm is an example of the converse:

It works in practice, can we make it work in theory?

Why is it important to learn why it works in theory?

1. Make sure it really works. This is low-priority. Hey! It works!
2. If we know how it works in theory then perhaps can improve it. This is high-priority. Commonly theory and practice work together to improve both.