Factoring Algorithms: Pollard
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 could them as $(\lg x)^2$ steps where $x$ is larger number you are dealing with.
- For RSA we want to factor $pq$ but our algs works for any $N$.

Multiplication HS Alg is $\lg x^2$ time. Tell Kolmogorov story.
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 $N^{1/2} = 2^{L/2}$.
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 $N^{1/2}$. 

More Concrete Goal: We present Pollard's algorithm which seems to be time $N^{1/4} = 2^{L/4}$.

Is this important? Discuss.

Yes since (1) perhaps we can build on this advance, (2) forces Alice and Bob to up their game.
Easy Factoring Algorithm

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

More Concrete Goal We present Pollard’s algorithm which seems to be time $N^{1/4} = 2^{L/4}$.

Is this important? Discuss.
Easy Factoring Algorithm

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

**More Concrete Goal** We present Pollard’s algorithm which seems to be time $N^{1/4} = 2^{L/4}$.

Is this important? Discuss.

**Yes** since (1) perhaps we can build on this advance, (2) forces Alice and Bob to up their game.
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 $N$.

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

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

We look at several approaches to finding such an $x, y$ that do not work before presenting the approach that does work.
Approach One: Random Sequence mod $p$

$x_1 = \text{RAND}(0,N-1)$
i=2
FOUND = FALSE
while NOT FOUND
    $x_i := \text{RAND}(0,N-1)$
    for j=1 to i-1
        if $x_i \equiv x_j \mod p$ then $d=\text{GCD}(x_i-x_j,N)$
        if (d NE 1) and (d NE N) then FOUND=TRUE
    i=i+1
output(d)
**Approach One: Random Sequence mod \( p \)**

\[
x_1 = \text{RAND}(0,N-1) \\
i = 2 \\
\text{FOUND} = \text{FALSE} \\
\text{while NOT FOUND} \\
\quad x_i := \text{RAND}(0,N-1) \\
\quad \text{for } j = 1 \text{ to } i-1 \\
\quad \quad \text{if } x_i \equiv x_j \pmod{p} \text{ then } d = \text{GCD}(x_i-x_j,N) \\
\quad \quad \text{if } (d \neq 1) \text{ and } (d \neq N) \text{ then } \text{FOUND}=\text{TRUE} \\
\quad \quad i = i + 1 \\
\text{output}(d)
\]

**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: Random Sequence mod $p$

\[ x_1 = \text{RAND}(0,N-1) \]
\[ i=2 \]
\[ \text{FOUND} = \text{FALSE} \]
while NOT FOUND
\[ x_i := \text{RAND}(0,N-1) \]
for j=1 to i-1
\[ \text{if } x_i \equiv x_j \pmod{p} \text{ then } d=\text{GCD}(x_i-x_j,N) \]
\[ \text{if } (d \neq 1) \text{ and } (d \neq N) \text{ then } \text{FOUND} = \text{TRUE} \]
i=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:** Don’t know $p$ so can’t run this algorithm.
Approach One: Random Sequence mod $p$

\[
\begin{align*}
x_1 &= \text{RAND}(0,N-1) \\
i &= 2 \\
\text{FOUND} &= \text{FALSE} \\
\text{while NOT FOUND} \\
&\quad x_i := \text{RAND}(0,N-1) \\
&\quad \text{for } j=1 \text{ to } i-1 \\
&\quad \quad \text{if } x_i \equiv x_j \pmod{p} \text{ then } d=\text{GCD}(x_i-x_j,N) \\
&\quad \quad \quad \text{if } (d \neq 1) \text{ and } (d \neq N) \text{ then } \text{FOUND}=\text{TRUE} \\
&\quad \quad \quad i = i+1 \\
\text{output}(d)
\end{align*}
\]

**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:** Don’t know \(p\) so can’t run this algorithm.

**ADJUST:** Always do GCD.
Approach One: Random Sequence mod $p$, Cont.

\[ x_1 = \text{RAND}(0,N-1) \]
\[ i=2 \]
\[ \text{FOUND} = \text{FALSE} \]
\[ \text{while NOT FOUND} \]
\[ \quad x_i := \text{RAND}(0,N-1) \]
\[ \quad \text{for } j=1 \text{ to } i-1 \]
\[ \quad \quad d=\text{GCD}(x_i-x_j,N); \text{ if } (d \text{ NE } 1 \text{ && } d \text{ NE } N) \text{ then } \text{FOUND=}\text{TRUE} \]
\[ \quad i=i+1 \]
\[ \text{output}(d) \]
Approach One: Random Sequence mod $p$, Cont.

\[ x_1 = \text{RAND}(0,N-1) \]
\[ i=2 \]
\[ \text{FOUND} = \text{FALSE} \]
\[ \text{while NOT FOUND} \]
  \[ x_i := \text{RAND}(0,N-1) \]
  \[ \text{for } j=1 \text{ to } i-1 \]
  \[ d=\text{GCD}(x_i-x_j,N); \text{ if } (d \neq 1 \text{ && } d \neq N) \text{ then } \text{FOUND=TRUE} \]
  \[ i=i+1 \]
\[ \text{output}(d) \]

PRO: Bday paradox: $x_i$’s:balls, mod $p$:boxes. Prob find $x_i \equiv x_j \pmod{p}$ with $i \leq p^{1/2} \sim N^{1/4}$. Sooner-other prime factors. Not knowing $p$ does not matter.
Approach One: Random Sequence mod $p$, Cont.

$x_1 = \text{RAND}(0,N-1)$

i=2

FOUND = FALSE

while NOT FOUND

\[ x_i := \text{RAND}(0,N-1) \]

for $j=1$ to $i-1$

\[ d = \text{GCD}(x_i-x_j,N); \text{ if } (d \neq 1 \&\& d \neq N) \text{ then FOUND=TRUE} \]

i=i+1

output(d)

**PRO:** Bday paradox: $x_i$’s:balls, mod $p$:boxes. Prob find $x_i \equiv x_j$ (mod $p$) with $i \leq p^{1/2} \sim N^{1/4}$. Sooner-other prime factors. Not knowing $p$ does not matter.

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

\[
\sum_{i=1}^{N^{1/4}} i^2 \sim (N^{1/4})^3 \sim N^{3/4} \text{ BAD} .
\]