Number-Theoretic Algorithms

Chapter 31, CLRS book
Modular Arithmetic
Integers

- $a | b$: $a$ divides $b$, $a$ is a divisor of $b$.
- $\text{gcd}(a,b)$: greatest common divisor of $a$ and $b$.
- Coprime or relatively prime: $\text{gcd}(a,b) = 1$.
- Euclid's algorithm: compute $\text{gcd}(a,b)$.
- Extended Euclid's algorithm: compute integers $x$ and $y$ such that $ax + by = \text{gcd}(a,b)$. 
Integers modulo $n$

- Let $n \geq 2$ be an integer.
- Def: $a$ is congruent to $b$ modulo $n$, written $a \equiv b \mod n$, if $n \mid (a - b)$, i.e., $a$ and $b$ have the same remainder when divided by $n$.
- Note: $a \equiv b \mod n$ and $a = b \mod n$ are different.
- Def: $[a]_n = \{\text{all integers congruent to } a \text{ modulo } n\}$.
- $[a]_n$ is called a residue class modulo $n$, and $a$ is a representative of that class.
- There are exactly $n$ residue classes modulo $n$:
  \[ [0], [1], [2], \ldots, [n-1]. \]
- Note: "congruence mod $n$" is an equivalence relation, whose equivalence classes are the residue classes.
- If \( x \in [a], y \in [b], \) then \( x + y \in [a + b] \) and \( x \cdot y \in [a \cdot b] \).
- Define addition and multiplication for residue classes:
  \[
  [a] \underbrace{+}_n [b] = [a + b] \\
  [a] \underbrace{\cdot}_n [b] = [a \cdot b].
  \]
Group

• A group, denoted by \((G,\ast)\), is a set \(G\) with a binary operation \(\ast : G \times G \to G\) such that
  1. \(\forall x, y \in G, \ x \ast y \in G\) (closure)
  2. \(x \ast (y \ast z) = (x \ast y) \ast z\) (associativity)
  3. \(\exists e \in G\) s.t. \(\forall x \in G, \ e \ast x = x \ast e = x\) (identity)
  4. \(\forall x \in G, \ \exists y \in G\) s.t. \(x \ast y = y \ast x = e\) (inverse)

• A group \((G,\ast)\) is abelian if \(\forall x, y \in G, \ x \ast y = y \ast x\).

• Examples: \((Z,+), (Q,+), (Q \setminus \{0\}, \times), (R,+), (R \setminus \{0\}, \times)\).
• Define $Z_n = \{[0], [1], ..., [n-1]\}$.

• Or, more conveniently, $Z_n = \{0, 1, ..., n-1\}$.

• $(Z_n, +)$ forms an abelian additive group.

• For $a, b \in Z_n$,
  - $a + b = (a + b) \mod n$. (Or, $[a] + [b] = [a + b] = [a + b \mod n]$.)
  - 0 is the identity element.
  - The inverse of $a$, denoted by $-a$, is $n - a$.

• When doing addition/subtraction in $Z_n$, just do the regular addition/subtraction and reduce the result modulo $n$.
  - In $Z_{10}$, $5 + 5 + 9 + 4 + 6 + 2 + 8 + 3 = ?$
• \((Z_n, *)\) is not a group, because \(0^{-1}\) does not exist.

• Even if we exclude 0 and consider only \(Z_n^+ = Z_n \setminus \{0\}\), \((Z_n^+, *)\) is not necessarily a group; some \(a^{-1}\) may not exist.

• For \(a \in Z_n\), \(a^{-1}\) exists if and only if \(\gcd(a, n) = 1\).
Let $Z_n^* = \{ a \in Z_n : \gcd(a, n) = 1 \}$.

$(Z_n, \ast)$ is an abelian multiplicative group.

$a \ast b = ab \mod n$.

$\ast = \ast \mod n$.

1 is the identity element.

The inverse of $a$, written $a^{-1}$, can be computed by the Extended Euclidean Algorithm.

For example, $Z_{12}^* = \{1, 5, 7, 11\}$. $5 \ast 7 = 35 \mod 12 = 11$.

Q: How many elements are there in $Z_n^*$?
• Euler's totient function:

\[ \varphi(n) = \left| \mathbb{Z}_n^* \right| \]

\[ = \left| \{a : 1 \leq a \leq n \text{ and } \gcd(a, n) = 1\} \right| \]

• Facts:

1. \( \varphi(p^e) = (p - 1)p^{e-1} \) for prime \( p \)

2. \( \varphi(ab) = \varphi(a) \varphi(b) \) if \( \gcd(a, b) = 1 \)
• Let $G$ be a (multiplicative) finite group.
• The order of $a \in G$, written $\text{ord}(a)$, is the smallest positive integer $t$ such that $a^t = e$. (\textit{e}, identity element.)
• Lagrange's theorem: For any element $a \in G$, $\text{ord}(a) | |G|$. 
• Corollary: For any element $a \in G$, $a^{|G|} = e$.
• Fermat's little theorem:
  If $a \in \mathbb{Z}_p^* \ (p \text{ a prime})$, then $a^{\phi(p)} = a^{p-1} = 1$ in $\mathbb{Z}_p^*$.
• Euler's theorem:
  If $a \in \mathbb{Z}_n^* \ (\text{for any } n > 1)$, then $a^{\phi(n)} = 1$ in $\mathbb{Z}_n^*$.
Example: $n = 15$

- $Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$
- $|Z_{15}^*| = \phi(15) = \phi(3) \times \phi(5) = 2 \times 4 = 8$

<table>
<thead>
<tr>
<th>$a \in Z_{15}^*$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>7</th>
<th>8</th>
<th>11</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>ord($a$)</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

- $a^{\phi(n)} = a^8 = 1$
Algorithms

- $\gcd(a, b)$
- $a^{-1} \mod n$
- $a^k \mod n$
- Running time: $O(\log^3 n)$
Euclidean Algorithm

Comment: compute $\gcd(a, b)$, where $a > b > 1$.

$$r_0 := a$$
$$r_1 := b$$

for $i := 1, 2, \ldots$ until $r_{n+1} = 0$

$$r_{i+1} := r_{i-1} \mod r_i$$

return $(r_n)$

Running time:

- $O(\log a)$ iterations; $O(\log^2 a)$ time for each mod.
- Overall running time: $O(\log^3 a)$
Extended Euclidean Algorithm

Given \( a > b > 0 \), compute \( x, y \) such that \( \gcd(a,b) = ax + by \).

Example: \( \gcd(299, 221) = ? \)

\[
299 = 1 \times 221 + 78 \\
221 = 2 \times 78 + 65 \\
78 = 1 \times 65 + 13 \\
65 = 5 \cdot 13 + 0
\]

\[
\gcd(229, 221) = 13 = 78 - 65 \\
= 78 - (221 - 2 \times 78) = 3 \cdot 78 - 221 \\
= 3 \times (299 - 1 \cdot 221) - 221 \\
= 3 \times 299 - 4 \times 221
\]
How to compute $a^{-1} \mod n$?

- Compute $a^{-1}$ in $\mathbb{Z}_n^*$.
- $a^{-1}$ exists if and only if $\gcd(a, n) = 1$.
- Use extended Euclidean algorithm to find $x, y$ such that $ax + ny = \gcd(a, n) = 1$ (in $\mathbb{Z}$)
  
  $\Rightarrow [a][x] + [n][y] = [1]$
  
  $\Rightarrow [a][x] = [1] \quad \text{ (since } [n] = [0])$
  
  $\Rightarrow [a]^{-1} = [x].$

- Note: may omit $[\ ]$, but reduce everything modulo $n$. 
Example

- Compute $15^{-1} \mod 47$.

  $47 = 15 \times 3 + 2$ (divide 47 by 15; remainder = 2)
  $15 = 2 \times 7 + 1$ (divide 15 by 2; remainder = 1)

  $1 = 15 - 2 \times 7$ (mod 47)
  $= 15 - (47 - 15 \times 3) \times 7$ (mod 47)
  $= 15 \times 22 - 47 \times 7$ (mod 47)
  $= 15 \times 22$ (mod 47)

  $15^{-1} \mod 47 = 22$

  That is, $15^{-1} = 22$ in $\mathbb{Z}_{47}^*$
Algorithm: Square-and-Multiply($x, c, n$)

Comment: compute $x^c \mod n$, where $c = c_k c_{k-1} \ldots c_0$ in binary.

$z \leftarrow 1$

for $i \leftarrow k$ downto 0 do

$z \leftarrow z^2 \mod n$

if $c_i = 1$ then $z \leftarrow (z \cdot x) \mod n$

return $(z)$

Note: At the end of iteration $i$, $z = x^{c_k \ldots c_i}$. 
Example: $11^{23} \mod 187$

$23 = 10111_b$

$z \leftarrow 1$

$z \leftarrow z^2 \cdot 11 \mod 187 = 11$ (square and multiply)

$z \leftarrow z^2 \mod 187 = 121$ (square)

$z \leftarrow z^2 \cdot 11 \mod 187 = 44$ (square and multiply)

$z \leftarrow z^2 \cdot 11 \mod 187 = 165$ (square and multiply)

$z \leftarrow z^2 \cdot 11 \mod 187 = 88$ (square and multiply)
The RSA Cryptosystem
The RSA Cryptosystem

- Best known and most widely used public-key scheme.
- Based on the assumed one-way property of modular powering:
  \[ f : x \rightarrow x^e \mod n \]  
  \[ f^{-1} : x^e \rightarrow x \mod n \]  
- In turn based on the hardness of integer factorization.
Idea behind RSA

It works in group \( \mathbb{Z}_n^* \). Let \( x \in \mathbb{Z}_n^* \) be a message.

Encryption (easy): \( x \xrightarrow{\text{RSA}} x^e \)

Decryption (hard): \( x \leftarrow x^e \xrightarrow{\text{RSA}^{-1}} \)

Looking for a "trapdoor": \( (x^e)^d = x \).

If \( d \) is a number such that \( ed \equiv 1 \pmod{\varphi(n)} \), then \( ed = k\varphi(n) + 1 \) for some \( k \), and

\[
(x^e)^d = x^{ed} = x^{\varphi(n)k+1} = (x^{\varphi(n)})^k \cdot x = 1 \cdot x = x.
\]
RSA Cryptosystem

- **Key generation:**
  
  (a) Choose large primes $p$ and $q$, and let $n := pq$.
  
  (b) Choose $e \ (1 < e < \varphi(n))$ coprime to $\varphi(n)$, and compute $d := e^{-1} \mod \varphi(n)$. ($ed \equiv 1 \mod \varphi(n)$.)
  
  (c) Public key: $pk = (n, e)$. Secret key: $sk = (n, d)$.

- **Encryption:** $E_{pk}(x) := x^e \mod n$, where $x \in \mathbb{Z}_n^*$.

- **Decryption:** $D_{sk}(y) := y^d \mod n$, where $y \in \mathbb{Z}_n^*$. 
Why RSA Works?

- The setting of RSA is the group \((Z_n^*, \cdot)\):
  - In group \((Z_n^*, \cdot)\), for any \(x \in Z_n^*\), we have \(x^{\phi(n)} = 1\).
  - We have chosen \(e, d\) such that \(ed \equiv 1 \mod \phi(n)\), i.e., \(ed = k\phi(n) + 1\) for some positive integer \(k\).
  - For \(x \in Z_n^*\), \((x^e)^d = x^{ed} = x^{k\phi(n)+1} = (x^{\phi(n)})^k x = x\).
RSA Example: Key Setup

- Select two primes: \( p = 17, \ q = 11 \).
- Compute the modulus \( n = pq = 187 \).
- Compute \( \varphi(n) = (p - 1)(q - 1) = 160 \).
- Select \( e \) between 0 and 160 such that \( \gcd(e, 160) = 1 \). Say \( e = 7 \).
- Compute \( d = e^{-1} \mod \varphi(n) = 7^{-1} \mod 160 = 23 \) (using extended Euclid's algorithm).
- Public key: \( pk = (e, \ n) = (7, \ 187) \).
- Secret key: \( sk = (d, \ n) = (23, \ 187) \).
RSA Example: Encryption & Decryption

• Suppose $m = 88$.
• Encryption: $c = m^e \mod n = 88^7 \mod 187 = 11$.
• Decryption: $m = c^d \mod n = 11^{23} \mod 187 = 88$.
• When computing $11^{23} \mod 187$, we do not first compute $11^{23}$ and then reduce it modulo 187.
• Rather, use square-and-multiply, and reduce intermediate results modulo 187 whenever they get bigger than 187.
Encryption Key $e$

- To speed up encryption, small values are usually used for $e$.

- Popular choices are $3$, $17 = 2^4 + 1$, $65537 = 2^{16} + 1$. These values have only two 1's in their binary representation.

- There is an interesting attack on small $e$. 
Attacks on RSA
Attacks on RSA

- Four categories of attacks on RSA:
  - brute-force key search
    (infeasible given the large key space)
  - mathematical attacks
  - timing attacks
  - chosen ciphertext attacks
Mathematical Attacks

- **Factor** \( n \) **into** \( pq \). Then \( \varphi(n) = (p - 1)(q - 1) \) and 
  \[ d = e^{-1} \mod \varphi(n) \] 
  can be calculated easily.

- **Determine** \( \varphi(n) \) **directly**. Equivalent to factoring \( n \). Knowing \( \varphi(n) \) will enable us to factor \( n \) by solving
  \[
  \begin{align*}
  n &= pq \\
  \varphi(n) &= (p - 1)(q - 1)
  \end{align*}
  \]

- **Determine** \( d \) **directly**. If \( d \) is known, \( n \) can be factored with high probability.
Integer Factorization

- A difficult problem, assumed to be infeasible.
- More and more efficient algorithms have been developed.
- In 1977, RSA challenged researchers to decode a ciphertext encrypted with a key \((n)\) of 129 digits (428 bits). Prize: $100. RSA thought it would take quadrillion years to break the code using fastest algorithms and computers of that time. Solved in 1994.
- In 1991, RSA put forward more challenges, with prizes, to encourage research on factorization.
RSA Numbers

- Each RSA number is a semiprime. (A number is semiprime if it is the product of two primes.)
- There are two labeling schemes.
  - by the number of decimal digits:
    RSA-100, ..., RSA-500, RSA-617.
  - by the number of bits:
    RSA-576, 640, 704, 768, 896, 1024, 1536, 2048.
RSA Numbers which have been factored

- RSA-100 (332 bits), 1991, 7 MIPS-year, Quadratic Sieve.
- RSA-110 (365 bits), 1992, 75 MIPS-year, QS.
- RSA-120 (398 bits), 1993, 830 MIPS-year, QS.
- RSA-129 (428 bits), 1994, 5000 MIPS-year, QS.
- RSA-130 (431 bits), 1996, 1000 MIPS-year, GNFS.
- RSA-140 (465 bits), 1999, 2000 MIPS-year, GNFS.
- RSA-155 (512 bits), 1999, 8000 MIPS-year, GNFS.
- RSA-160 (530 bits), 2003, Lattice Sieve.
- RSA-200 (663 bits), 2005, Lattice Sieve.
RSA-200 =
27,997,833,911,221,327,870,829,467,638,
722,601,621,070,446,786,955,428,537,560,
009,929,326,128,400,107,609,345,671,052,
955,360,856,061,822,351,910,951,365,788,
637,105,954,482,006,576,775,098,580,557,
613,579,098,734,950,144,178,863,178,946,
295,187,237,869,221,823,983.
Remarks

- In light of current factorization technologies, RSA recommends \(|n| = 1024-2048\) bits.

- If a message \(m \in \mathbb{Z}_n \setminus \mathbb{Z}_n^*\),
  - RSA works, but
  - Since \(\gcd(m, n) > 1\), the sender can factor \(n\).
  - Since \(\gcd(m^e, n) > 1\), the adversary can factor \(n\), too.

- Question: how likely is \(m \in \mathbb{Z}_n \setminus \mathbb{Z}_n^*\)?
Generating large primes

To set up an RSA cryptosystem, we need two large primes $p$ and $q$. 
How many prime numbers are there?

- Infinitely many.
- First proved by Euclid:
  - Assume only a finite number of primes $p_1, p_2, \ldots, p_n$.
  - Let $M = p_1 p_2 \ldots p_n + 1$.
  - $M$ is not a prime, because $M \neq p_i$, $1 \leq i \leq n$.
  - So, $M$ is composite and has a prime factor $p_i$ for some $i$
    \[ \Rightarrow p_i \mid M \Rightarrow p_i \mid 1 \Rightarrow \text{contradiction}. \]
Distribution of Prime Numbers

The Prime Number Theorem:
Let \( \pi(x) \) denote the number of primes \( \leq x \). Then
\[
\pi(x) \approx \frac{x}{\ln x} \quad \text{for large } x.
\]

Dirichlet's Theorem: For \( b \in \mathbb{Z}_n^* \), let \( \pi_{n,b}(x) \) denote the number of primes \( y \) such that \( y \leq x \) and \( y \equiv b \mod n \). Then,
\[
\pi_{n,b}(x) \approx \frac{x}{\ln x} \cdot \frac{1}{\varphi(n)} \quad \text{for large } x.
\]
How to generate a large prime number?

- Generate a random odd number $n$ of desired size.
- Test if $n$ is prime.
- If not, discard it and try a different number.
- Q: How many numbers are expected to be tested before a prime is found?
Primality test: Is $n$ a prime?

- Can it be solved in polynomial time?
- A long standing open problem until 2002.
- AKS (Agrawal, Kayal, Saxena): $O\left(\left(\log n\right)^{12+\varepsilon}\right)$.
  - Later improved by others to $O\left(\left(\log n\right)^{10.5}\right)$, and then to $O\left(\left(\log n\right)^{6+\varepsilon}\right)$.
- In practice, Miller-Rabin's probabilistic algorithm is still the most popular --- much faster, $O\left(\left(\log n\right)^{3}\right)$.
Miller-Rabin primality test: Is $n$ a prime?

- Looking for a characteristic property of prime numbers:
  - $n$ is prime $\iff$ what?
  - $n$ is prime $\iff \forall a \in \mathbb{Z}_n^*, P(a) = \text{true}$
  - $n$ is prime $\Rightarrow \forall a \in \mathbb{Z}_n^*, P(a) = \text{true}$
    
    not prime $\Rightarrow \exists k$ elements $a \in \mathbb{Z}_n^*, P(a) = \text{false}$

- Check $P(a)$ for $t$ random elements $a \in \mathbb{Z}_n^*$.
  - If $P(a)$ all true, then return "prime"
    
    else return "composite."
  - A "prime" answer may be incorrect with prob $p(k,t)$.
  - If $k \geq \frac{1}{2} |\mathbb{Z}_n^*|$, then $p(k,t) \leq \frac{1}{2^t}$. 
If $n$ is prime, then for all $a \in \mathbb{Z}_n^*$, $P(a)$ is true.
If $n$ is not prime, then there are strong witnesses, which are elements $a \in \mathbb{Z}_n^*$ s.t. $P(a) = false$. 
• Looking for $P(a)$:
  • How about $P(a) = \left[ a^{n-1} \equiv 1 \mod n \right]$?

• Fermat's little theorem:
  If $n$ is prime $\Rightarrow \forall a \in \mathbb{Z}_n^*, a^{n-1} \equiv 1 \mod n$.

• If $n$ is not prime $\Rightarrow$ maybe no strong witnesses.

  (Carmichael numbers: composite numbers $n$
  for which $a^{n-1} \equiv 1 \mod n \ \forall a \in \mathbb{Z}_n^*$.)

• Need to refine the condition $\left[ a^{n-1} \equiv 1 \mod n \right]$. 
• Fact: if \( n \neq 2 \) is prime, then 1 has exactly two square roots in \( \mathbb{Z}_n^* \), namely \( \pm 1 \).

• Write \( n - 1 = u2^k \), where \( u \) is odd.

• If \( n \) is prime

\[
\Rightarrow \quad \forall a \in \mathbb{Z}_n^*, \quad a^{u2^k} \equiv 1 \mod n \quad \text{(Fermat's little theorem)}
\]

\[
\Rightarrow \quad \forall a \in \mathbb{Z}_n^*, \ P(a) = true, \text{ where }
\]

\[
P(a) = \begin{cases} 
    a^u \equiv 1 \mod n \text{ or } \\
    a^{u2^i} \equiv -1 \mod n \text{ for some } i, \ 0 \leq i \leq k - 1
\end{cases}
\]

• Why? Consider the sequence

\[
a^u, \ a^{u2}, \ a^{u2^2}, \ldots, \ a^{u2^{k-1}}, \ a^{u2^k} = 1
\]
• If $n$ not prime $\Rightarrow$ do strong witnesses always exist?

• Loosely speaking, yes: if $n$ is an odd composite and not a prime power, then at least one half of the elements $a \in Z_n^*$ are strong witnesses.

• A composite number $n$ is a prime power if $n = p^e$ for some prime $p$ and integer $e \geq 2$. (A perfect power if $n = k^e$ for some integer $k$ and $e \geq 2$.)
• **Theorem:** If $n$ is an odd composite and not a prime power, then at least one half of the elements $a \in \mathbb{Z}_n^*$ are strong witnesses.

• **Sketch of proof:** The set $A$ of *non*-strong witnesses forms a proper subgroup of $\mathbb{Z}_n^*$. So, $\text{ord}(A) < \text{ord}(\mathbb{Z}_n^*)$ and $\text{ord}(A) | \text{ord}(\mathbb{Z}_n^*)$. So, $\text{ord}(A) \leq \frac{1}{2} \text{ord}(\mathbb{Z}_n^*)$. 
Algorithm: Miller-Rabin primality test

- Input: integer $n > 2$ and parameter $t$
- Output: a decision as to whether $n$ is prime or composite

1. if $n$ is even, return "composite"
2. if $n$ is a perfect power, return "composite"
3. for $i := 1$ to $t$ do
   - choose a random integer $a$, $2 \leq a \leq n-1$
   - if $\gcd(a, n) \neq 1$, return "composite"
   - if $a$ is a strong witness, return "composite"
4. return ("prime")
Analysis: Miller-Rabin primality test

- If the algorithm answers "composite", it is always correct.
- If the algorithm answers "prime", it may or may not be correct.
- The algorithm gives a wrong answer if $n$ is composite but the algorithm fails to find a strong witness in $t$ iterations.
- This may happen with probability at most $2^{-t}$.
- Actually, at most $4^{-t}$, by a more sophisticated analysis.
Monte Carlo algorithms

• A Monte Carlo algorithm is a probabilistic algorithm
  • which always gives an answer
  • but sometimes the answer may be incorrect.

• A Monte Carlo algorithm for a decision problem is yes-biased if its “yes” answer is always correct but a “no” answer may be incorrect with some error probability.

• A $t$-iteration Miller-Rabin is a “composite”-biased Monte Carlo algorithm with error probability at most $1/4^t$. 
Las Vegas algorithms

- A Las Vegas algorithm is a probabilistic algorithm
  - which may sometimes fail to give an answer
  - but never gives an incorrect one
- A Las Vegas algorithm can be converted into a Monte Carlo algorithm.
Integer Factorization

Reference on quadratic sieve:

http://blogs.msdn.com/b/devdev/archive/2006/06/19/637332.aspx
Fermat's Method

- **Difference of squares**
  - To factor \( n \), find an \( a > n \) such that \( a^2 - n \) is a square, say \( b^2 \).
  - Then, \( n = a^2 - b^2 = (a - b)(a + b) \).
  - Search for \( a \) starting from \( a = \left\lceil \sqrt{n} \right\rceil \).

- **Example:** Suppose \( n = 5959 \). Then, \( \left\lceil \sqrt{n} \right\rceil = 78 \).
  - \( a^2 - n \) is not a square for \( a = 78 \) and 79.
  - \( a^2 - n \) is a square for \( a = 80 \): \( 80^2 - 5959 = 441 = 21^2 \).
  - Hence \( 5959 = 80^2 - 21^2 = (80 - 21)(80 + 21) = 59 \times 101 \).
  - **Slow:** a linear search for \( b^2 = a^2 - n \) is a poor strategy.
Dixon's Random Squares Algorithm

- Basic idea: a generation of Fermat's difference of squares.
  - To factor $n$, find $x \not\equiv \pm y \pmod{n}$ such that $x^2 \equiv y^2 \pmod{n}$.
  - Then, $n \mid (x - y)(x + y)$, but $n$ divides neither of $x \pm y$.
  - Hence, $\gcd(x \pm y, n)$ are nontrivial factors of $n$.

- Example: $32^2 \equiv 10^2 \pmod{77}$. $\gcd(32 \pm 10, 77) = 7$ and $11$.

- Question: how to produce such $x$ and $y$?

- Factor base: a set $B$ of small primes, say, $B = \{p_1, p_2, \ldots p_b\}$.

- An integer $z$ is smooth if it can be factored over $B \mod n$, i.e., $z \equiv p_1^{e_1} p_2^{e_2} \ldots p_b^{e_b} \mod n$ for some $e_1, e_2, \ldots, e_b \geq 0$. 
• Our goals:
  • First, find a set $U$ of integers $x_i$ such that $x_i^2$ are smooth:
    \[ x_i^2 \equiv p_1^{e_{i1}} p_2^{e_{i2}} \ldots p_b^{e_{ib}} \mod n \]
  • Second, select a subset $S \subseteq U$ such that the product
    \[ \prod_{x_i \in S} x_i^2 \] has an even exponent for each $p_i$, say,
    \[ \prod_{x_i \in S} x_i^2 \equiv p_1^{2e_1} p_2^{2e_2} \ldots p_b^{2e_b} \mod n \] for some $e_1, e_2, \ldots, e_b \geq 0$.

• Let $X = \prod_{x_i \in S} x_i \mod n$ and $Y = p_1^{e_1} p_2^{e_2} \ldots p_b^{e_b} \mod n$, and
  
  we have $X^2 \equiv Y^2 \mod n$.

• If $X \equiv \pm Y \mod n$, no luck, try a different set of $x_i$'s.
Example (from Stinson's book on Cryptography)

- Suppose \( n = 15770708441 \) and \( B = \{2, 3, 5, 7, 11, 13\} \).
- Consider the three congruences:
  \[
  8340934156^2 \equiv 3 \times 7 \mod n \\
  12044942944^2 \equiv 2 \times 7 \times 13 \mod n \\
  2773700011^2 \equiv 2 \times 3 \times 13 \mod n.
  \]
- \[
  (8340934156 \times 12044942944 \times 2773700011)^2 \\
  \equiv (2 \times 3 \times 7 \times 13)^2 \mod n.
  \]
- Reducing by modulo \( n \) yields \( (9503435785)^2 \equiv (546)^2 \mod n \).
- A factor of \( n \): \( \gcd(9503435785 - 546, 15770708441) = 115759 \).
To achieve our second goal

- Suppose \( B = \{p_1, p_2, \ldots p_b\} \). Let \( c > b \).

- Suppose we have a set \( U \) of \( c \) integers \( x_i \) such that \( x_i^2 \) are smooth:
  \[
x_i^2 \equiv p_1^{e_{i1}} p_2^{e_{i2}} \ldots p_b^{e_{ib}} \mod n \quad (1 \leq i \leq c).
  \]

- Let \( e_i = (e_{i1} \mod 2, e_{i2} \mod 2, \ldots, e_{ib} \mod 2) \).

- The \( c \) vectors \( e_i \) are linearly dependent (because \( c > b \)), and we can find a subset \( S \) of \( e_i \)'s that sum modulo 2 to \((0, 0, \ldots, 0)\).

- Let \( X = \prod x_i \mod n \) be the product of the \( x_i \)'s corresponding to the \( e_i \)'s in \( S \).
Example (cont.)

- We have $B = \{2, 3, 5, 7, 11, 13\}$ and
  
  $x_1^2 = 8340934156^2 \equiv 3 \times 7 \mod n$
  
  $x_2^2 = 12044942944^2 \equiv 2 \times 7 \times 13 \mod n$
  
  $x_3^2 = 2773700011^2 \equiv 2 \times 3 \times 13 \mod n.$

- $e_1 = (0, 1, 0, 1, 0, 0)$
  
  $e_2 = (1, 0, 0, 1, 0, 1)$
  
  $e_3 = (1, 1, 0, 0, 0, 1)$

- $e_1 + e_2 + e_3 \equiv (0, 0, 0, 0, 0, 0) \mod 2.$

- Thus, we let $X^2 = (x_1 x_2 x_3)^2 \mod n$ and
  
  $Y^2 = (3 \times 7)(2 \times 7 \times 13)(2 \times 3 \times 13) \mod n.$
Searching for smooth squares $x_i^2$

- Dixon's strategy: choose $x_i$ at random, hence the name Random Squares Method.

- Trick 1: try numbers of the form $x = j + \left\lfloor \sqrt{kn} \right\rfloor$, $j = 0, 1, 2, \ldots$, and $k = 1, 2, \ldots$. For such $x$, $x^2 \mod n$ tends to be small and has a better chance than average to be smooth.

- Trick 2: also try numbers of the form $x = \left\lfloor \sqrt{kn} \right\rfloor - j$, $j = 0, 1, 2, \ldots$, and $k = 1, 2, \ldots$. For such $x$, $x^2 \mod n$ is a little bit smaller than $n$. Try to factor $(x^2 \mod n) - n$ instead of $x^2 \mod n$.

- Trick 3: to play trick 2, we need to include $-1$ in $B$. 
Example (from Stinson's book on Cryptography)

- Suppose $n = 1829$ and $B = \{-1, 2, 3, 5, 7, 11, 13\}$.
- $\sqrt{n} = 42.77, \quad \sqrt{2n} = 60.48, \quad \sqrt{3n} = 74.07, \quad \sqrt{4n} = 85.53$.
- Thus we try $x = 42, 43, 60, 61, 74, 75, 85, 86$, and obtain

\[
\begin{align*}
x_1^2 & \equiv 42^2 \equiv -65 \equiv (-1) \times 5 \times 13. & e_1 & = (1, 0, 0, 1, 0, 0, 1) \\
x_2^2 & \equiv 43^2 \equiv 20 \equiv 2^2 \times 5. & e_2 & = (0, 0, 0, 1, 0, 0, 0) \\
x_3^2 & \equiv 61^2 \equiv 63 \equiv 3^2 \times 7. & e_3 & = (0, 0, 0, 0, 1, 0, 0) \\
x_4^2 & \equiv 74^2 \equiv -11 \equiv (-1) \times 11. & e_4 & = (1, 0, 0, 0, 0, 1, 0) \\
x_5^2 & \equiv 85^2 \equiv -91 \equiv (-1) \times 7 \times 13. & e_5 & = (1, 0, 0, 0, 1, 0, 1) \\
x_6^2 & \equiv 86^2 \equiv 80 \equiv 2^4 \times 5. & e_6 & = (0, 0, 0, 1, 0, 0, 0)
\end{align*}
\]
\begin{itemize}
  \item $e_2 + e_6 = (0,0,0,0,0,0)$, but does not yield a factorization of $n$.
  \item $(43 \times 86)^2 \equiv (2^3 \times 5)^2 \mod 1829$.
  \item $(3698)^2 \equiv (40)^2 \mod 1829$.
  \item $(40)^2 \equiv (40)^2 \mod 1829$.
  \item $e_1 + e_2 + e_3 + e_5 = (0,0,0,0,0,0)$.
  \item $(42 \times 43 \times 61 \times 85)^2 \equiv (-1 \times 2 \times 3 \times 5 \times 7 \times 13)^2 \mod 1829$.
  \item $1459^2 \equiv 901^2 \mod 1829$.
  \item $\gcd(1459 - 901, 1829) = 31$.
  \item $1829 = 31 \times 59$.
\end{itemize}
Quadratic Sieve

• Consider the interval \([M_1, M_2]\) around \(\sqrt{n}\) for some suitable integers \(M_1, M_2\).

• Let \(Q(x) = x^2 - n\). We want to find a set \(U\) of integers \(x\) for which \(Q(x)\) is smooth.

• Recall the factor base \(B = \{p_1, p_2, \ldots, p_b\}\).

• Recall Dixon's method (pick an \(x \in [M_1, M_2]\) and test if \(Q(x)\) is smooth) and observe how the computing time is wasted.

• Idea of QS: use each \(p \in B\) as a "sieve" and sieve it through \(A\).

• Notice that if \(p \in B\), \(x, y \in [M_1, M_2]\), and \(p \mid Q(x)\), then we have \(p \mid Q(y)\) iff \(x \equiv y \text{ mod } p\).
Sketch of the Quadratic Sieve Algorithm

1. Array $QA[M_1..M_2]$. Initially, $QA[i] \leftarrow i^2 - n$.

2. for each $p \leftarrow p_1, p_2, \ldots, p_b \in B$ do
   • find an $i \in [M_1..M_2]$ such that $p \mid Q(i)$;
   • for each $j \in [M_1..M_2]$ such that $i \equiv j \mod p$ do
     $QA[i] \leftarrow QA[i]/p^{e_i}$, where $e_i$ is the largest possible;
     keep record of $e_i \mod 2$.

3. Let $U$ be the set of all $i \in [M_1..M_2]$ such that $QA[i] = 1$.
   // $Q(i)$ is smooth for each $i \in S$ //

4. Construct a subset $S \subseteq U$ as in Dixon's.