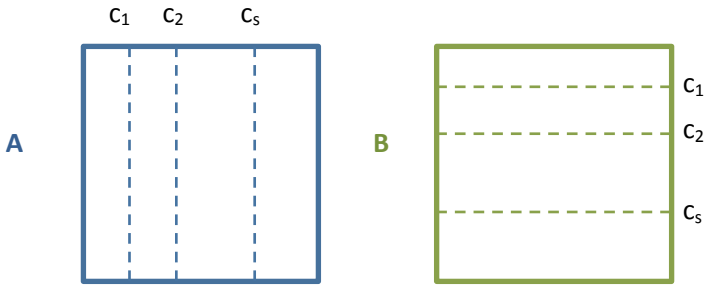
# CSE 6512 -- Lecture 15

*Notes by Son Le*

## Approximate Matrix multiplication




$$\boldsymbol{C = A \times B}$$

For all $(i, j)$, to compute $C_{ij}$,

- Pick $s$ columns randomly from matrix $A$, let them be $c_1, c_2, \dots, c_s$.
- Output $C_{ij} = \frac{n}{s}\sum_{q=1}^{s} A_{i,c_q} B_{c_q,j}$

Consider the case of A and B being Boolean. Note that C is not Boolean, for example:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 0 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 3 \end{bmatrix}$$

If $C_{ij} = q$ then there are $q$ matching 1's in row $i$ of $A$ and column $j$ of $B$. If we pick $s$ columns randomly from row $i$ of A, the expected number of these columns that are among the above $q$ columns is $\frac{sq}{n}$. Using Chernoff bounds, the number of matching columns in the sample is in the range $\left[\frac{sq}{n} - c\alpha\sqrt{\frac{sq}{n}\log n},\ \frac{sq}{n} + c\alpha\sqrt{\frac{sq}{n}\log n}\right]$ with probability $\geq 1 - n^{-\alpha}$, $c$ being a constant.

The error in the output is in the range $\left[-c\alpha\sqrt{\frac{nq}{s}\log n},\ c\alpha\sqrt{\frac{nq}{s}\log n}\right]$ with probability $\geq 1 - n^{-\alpha}$.

The runtime of this algorithm is $O(n^2 s)$.

# Primality testing

- **Input:** an integer $n$
- **Output:** *yes* if $n$ is prime; *no* otherwise.

**Fact:**

A trivial algorithm takes $O(\sqrt{n})$ time.

**Fermat's theorem (1640):**

If $p$ is prime then $a^{p-1} \equiv 1 \pmod{p}$ for any $a < p$.

Note that this is not an if-and-only-if statement. There are composite numbers (e.g., Carmichael numbers) for which the above holds for every $a < p$. An example Carmichael number is 561.

# Miller-Rabin's algorithm

## Computing $x^n$

Let $n = \overline{b_k b_{k-1} \dots b_1 b_0}$, where $b_i \in \{0, 1\}$.

$$x^n = x^{\sum_{i=0}^{k} b_i 2^i} = \prod_{i=0}^{k} x^{b_i 2^i} = \prod_{\substack{i=0 \\ b_i=1}}^{k} x^{2^i}$$

## The algorithm

**Witness($a$, $n$)**

*$a$ is the random number that we pick and $n$ is the input integer.*
Let $n - 1 = \overline{b_k b_{k-1} \dots b_1 b_0}$
$result \coloneqq 1;$
For $q \coloneqq k$ downto 0 do
    $y \coloneqq result;$
    $result \coloneqq (result * result) \bmod n;$
    If $result = 1, y \neq 1, y \neq n - 1$ then
        Return true;
    If $b_q = 1$ then $result \coloneqq (result * a) \bmod n;$
Return $result \neq 1$

Note that $\mathbb{Z}_n$ can have only two solutions for $x$ in $x^2 \bmod n = 1$ if $n$ is prime, since $\mathbb{Z}_n$ will be a field when $n$ is a prime.

**Check_if_prime($n$)**

For $i \coloneqq 1$ to $\alpha \log n$ do
    Pick a random integer $a < n$;
    If witness($a$, $n$) then output "$n$ is composite" and quit;
Output "$n$ is prime"

**Theorem:**

For any composite integer $n$, the number of witnesses is $\geq \frac{n-1}{2}$.

Therefore, if $n$ is composite, the probability that a random number $a < n$ is not a witness is $\leq \frac{1}{2}$. The probability that none of the $\alpha \log\ n$ chosen integers is a witness is $\leq \left(\frac{1}{2}\right)^{\alpha \log n} = n^{-\alpha}$.

The probability of an incorrect answer of Check_if_prime is $\leq n^{-\alpha}$.

The runtime of Check_if_prime is $O(\log^2 n)$.

Note that if the input integer is a prime, then the above algorithm will never give an incorrect output. Thus this algorithm has one-sided error.