Computational Complexity

Fabio A. González
QCP 2020-2

Universidad Nacional de Colombia

1. The analysis of computational problems

- what is a computational problem?

- How to design algorithms to solve particular computational problems?

Analysis and Design of Algorithms

  - Is the algorithm correct? (effectiveness)
    testing, formal methods
  - How much resources the algorithm requires? (efficiency)
    time, space, energy
  - which design alternatives do we have?
    brute force, greedy, dynamic progr. etc.

- what are the minimal resources we require to solve a computational problem?

Computational Complexity theory

2. How to quantify computational resources

```
                def insertion_sort(A):
O(1) {          j = 1
                while j < len(A):
O(n^2) { O(1) {     key = A[j]
                    i = j - 1
         O(j) {     while (i >= 0) and (A[i] > key):
                        A[i + 1] = A[i]
                        i = i -1
         O(1) {     A[i + 1] = key
                    j = j + 1
```

$\boxed{\text{Time}(n) \in O(n^2)}$

$O(n^2) = \sum_{j=1}^{n-1} \left(O(1) + O(j)\right)$

- Counting the type and number of instructions which are executed
- This depends on the input
  - length (n)
  - Order:
    - Worst case ✓
    - Best case
    - Average case
- Time(n): Amount of time expended by the algorithm with an input of size n in the worst case

  Space(n)

$O \leftrightarrow \leq \qquad \Omega \leftrightarrow \geq \qquad \Theta \leftrightarrow =$

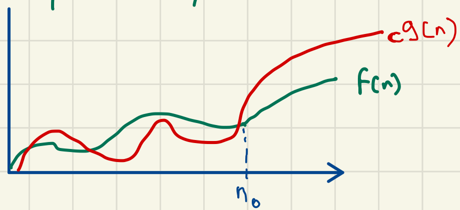
- Characterize the behavior of time(n) how it grows with n.

- Asymptotic notation

$O(g(n)) = \{ f(n) \mid \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \ \forall n \geq n_0 \ f(n) \leq c g(n) \}$

$f(n) \in O(g(n))$

$f(n) = O(g(n))$



$a_d n^d + a_{d-1} n^{d-1} \dots a_0 n^0 \in O(n^d)$

$\log n \in O(n)$

$n \log n \in O(n^2)$

$n \log n + n^2 \in O(n^2)$

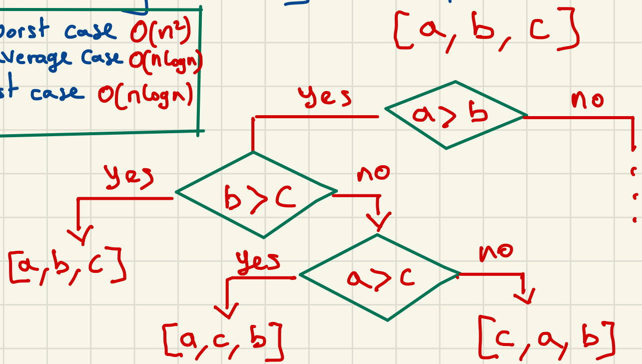
3. Computational complexity

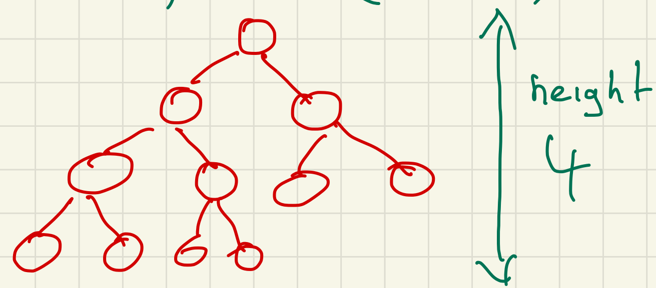
- It is the study of the time and space resources required to solve computational problems.

- what's the time of the best algorithm to sort?

Sorting using Comparisons $\Longrightarrow$ $\text{Time}(n) = \Omega(n \log n)$

Quicksort :	worst case $O(n^2)$ average case $O(n \log n)$
Heapsort :	worst case $O(n \log n)$





what's the minimum number of comparisons I have to do to sort an array of n elements?

- \# permutations $n!$ = \# leaves of the tree
- I want the tree with minimum height $\Longrightarrow$ balanced tree
- what's the minimum height of a binary tree with n leaves? $\Longrightarrow$ $\log_2 n$
- " " " " " " " " " $n!$ " ? $\Longrightarrow$ $\log_2(n!) \in \Omega(n \log n)$

4. Decision problems

- It is a problem where the answer is yes or no.
- A language L over the alphabet Σ is a subset of Σ^* (all finite strings over Σ).

  if $\Sigma = \{0, 1\}$ $\quad \Sigma^* = \{0, 1, 10, 01, \dots\}$

  $L = \{0, 10, 100, 110, \dots\}$ (even numbers)

- Decision problems can be encoded as languages:

  Primality problem $\longleftrightarrow$ $L \subseteq \{0,1\}^*$ $\quad L = \{n \mid n \text{ represents a prime number}\}$

- A language L is decided by a Turing machine if the machine is able to decide if an input on its tape belongs to the language or not.

  Decision $\to$ halts in $q_y \to$ yes

  Decision $\to$ halts in $q_n \to$ no

- We say that a problem (language (L)) is in Time$(f(n))$ if there exists a Turing Machine which decides L in time $O(f(n))$ (where n is the size of the input)

5. P and NP

- P = The collection of languages that can be decided in $\mathrm{Time}(n^k)$ for some $k \in \mathbb{N}$

Factoring problem : Given an integer m and $l < m$, does m have a non-trivial factor less than l.

witness : a number $1 < x < l$ such that x divides m

NP = The collection of problems/languages (L) such that there is a Turing machine M :

1) if $x \in L$ there exist a witness w such that M halts in q_y after a time polynomial in $|x|$ with an input "x_w"
2) if $x \notin L$ for all the candidate witnesses w the machine halts in q_n after a time polynom. in $|x|$ with an input "x_w"

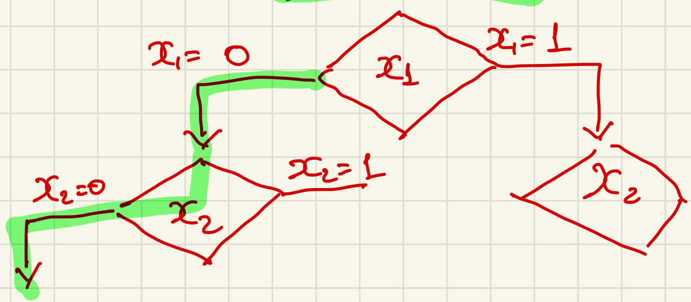
NP $\equiv$ Non-deterministic Polynomial

- **Sat**: Given a propositional formula F (e.g. $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$) is there a set of values for $x_1, x_2 \dots x_n$ such that $F(x_1 \dots x_n) = \text{True}$

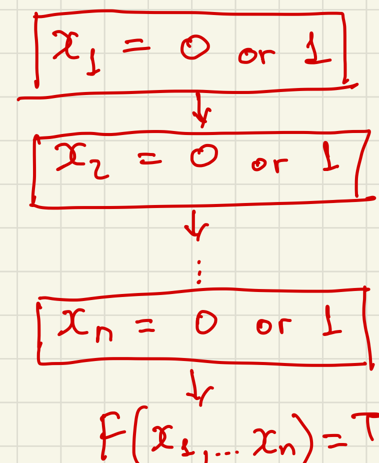
- Sat $\in$ NP? yes. witness: set of values for $x_1, \dots x_n$

Non-deterministic Turing Machine

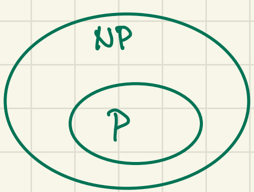
Deterministic



Non_deterministic

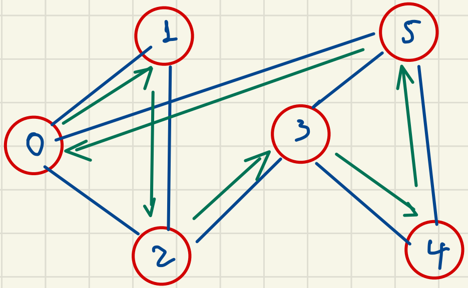


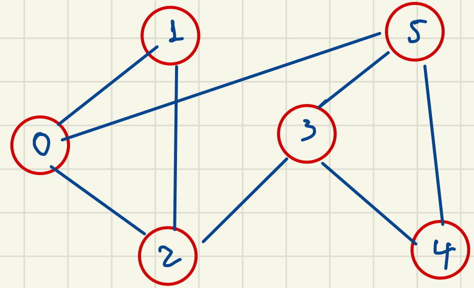
P = NP? Can NP problem be solved in polynomial time?



we don't know!

6. Hamiltonian and Euler cycle problem

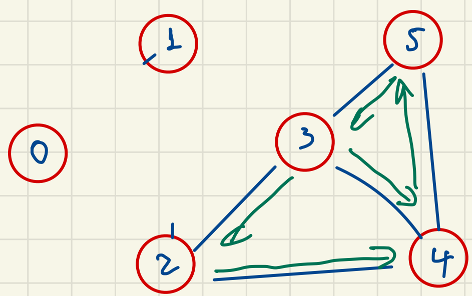




Euler Theorem:
A connected graph has an Euler cycle if and only if every vertex has an even number of edges incident over it

Hamiltonian cycle:
Is a sequence of vertices $v_1, v_2 \dots v_m v_1$ with (v_i, v_{i+1}) an edge such that it visits all the vertices exactly once (except for the first and last vertex)

Euler cycle:
Is a sequence of vertices $v_1, v_2 \dots v_m$ with (v_i, v_{i+1}) an edge such that it visits all the edges exactly once.



7. Problem Reduction

- A language B is said to be reducible to another language A if there is a TM operating in polynomial time that given an input x it outputs $R(x)$ and $x \in B \iff R(x) \in A$.

- A problem P is complete with respect to a complexity class if every language in the complexity class can be reduced to P.

- NP-Complete is the set of problems which are complete with respect to NP.

Does $x \in B$

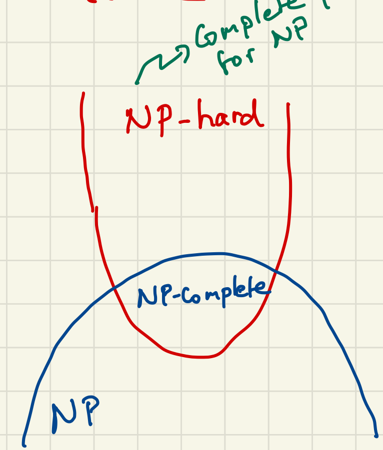
$\boxed{\text{Compute } R(x)}$ } Polynomial Time

$\downarrow$

$\boxed{R(x) \in A}$

$\downarrow$

yes or not



8. NP- Complete problems

- SAT
- 3-SAT (Propositions wit terms with at most 3 variable)
- CSAT (satisfactbility of boolean circuits)
- CLIQUE
- Vertex Cover
- Hamiltonian Cycle
- 0-1 integer programming
- TSP : Travel salesman problem.

8. Hierarchy of Complexity Classes

strict

$$L \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$$

strict

Logarithmic Space

Polynomial Space

Exponential Time

$A, B \in P$ Does $x \in B$

$x_1 \in A$

$x_2 \notin A$

$\boxed{\text{Compute } R(x)}$

$\downarrow$

$\boxed{R(x) \in A}$

$\downarrow$

yes or not

$$R(x) = \begin{cases} x_1 & \text{if } x \in B \\ x_2 & \text{if } x \notin B \end{cases}$$

$O(n^k)$

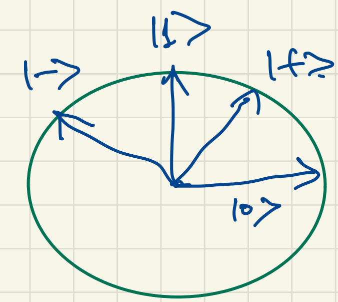
$O(n^k)$

$$|\psi\rangle = \alpha|v_0\rangle + \beta|v_1\rangle$$

$$= \cos\frac{\theta}{2}|v_0\rangle + \sin\frac{\theta}{2}e^{i\gamma}|v_1\rangle.$$

$$\operatorname{Spread}_{v_0,v_1}(|\psi\rangle) = \left|\cos\frac{\theta}{2}\right| + \left|\sin\frac{\theta}{2}\right| \leq \sqrt{2}$$

$$\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}$$



$$|0\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right)$$

$$\underbrace{|\psi\rangle = |0\rangle}$$

$$\begin{matrix}|+\rangle \\ |-\rangle\end{matrix} = \frac{1}{\sqrt{2}}\left(|0\rangle \pm |1\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(|\psi\rangle \pm |\psi^\perp\rangle\right)$$

$$\mathcal{U} \qquad \mathcal{U}\mathcal{U}^\dagger = \mathcal{U}^\dagger \mathcal{U} = \mathbb{1}$$

$$\boxed{\mathcal{U}|\psi\rangle = \underline{a}|\psi\rangle} \quad \text{t.q.} \quad \underline{|a| = 1}$$

$$\mathcal{U} = \lambda_1 |v_1\rangle\langle v_1| + \lambda_2 |v_2\rangle\langle v_2|$$

$$|\lambda_1| = |\lambda_2| = 1$$

$$\mathcal{U}|v_1\rangle = \lambda_1 |v_1\rangle$$

$$|q_1\, q_2 \cdots q_i \cdots q_n\rangle = \begin{matrix} |00\ldots0\rangle \\ |00\ldots1\rangle \\ \vdots \\ |11\ldots0\rangle \\ |11\ldots1\rangle \end{matrix} \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \vdots \\ \alpha_{11\ldots1} \end{pmatrix} \qquad \begin{matrix} |00\ldots0\rangle \\ |10\ldots0\rangle \\ \vdots \end{matrix}$$

$$q_i = 1$$

$$|q_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$$

$$|q_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$$

$$|q_3\rangle = \alpha_3|0\rangle + \beta_3|1\rangle$$

$$|q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$$
$$+$$
$$|q_1'\rangle \otimes |q_2'\rangle \otimes |q_3'\rangle$$

$$|\psi\rangle \longrightarrow |\psi\rangle\langle\psi| = \rho$$

1. Incertidumbre cuántica. Superposición.
2. Incertidumbre clásica.

$$\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}$$

$$\left.\begin{matrix} |\psi_1\rangle & P_1 \\ |\psi_2\rangle & P_2 \\ \vdots & \\ |\psi_n\rangle & P_n \end{matrix}\right] \rho$$

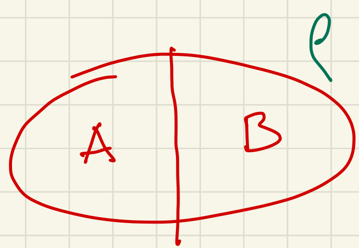
$$\rho = \sum_{i=1}^{n} P_i |\psi_i\rangle\langle\psi_n|$$

$$\mathcal{E}(\rho) = tr(\rho \log_2 \rho)$$

$$\mathcal{E}(\rho) = 0 \Rightarrow \text{Pure}$$

$$\mathcal{E}(\rho) \neq 0 \Rightarrow \text{Mixed}$$

$$tr\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix} = \sum_{i=1}^{n} a_{ii}$$



$$\rho_A = tr_B(\rho)$$

$$\rho_B = tr_A(\rho)$$

if ρ_A is Mixed or ρ_B is Mixed then A is Entangled with B

$$\mathrm{tr}_A(A \otimes B) = \begin{bmatrix} \boxed{a_{11}[B]} & a_{12}[B] & \\ \vdots & & \\ a_{n1}[B] & & \boxed{a_{nn}[B]} \end{bmatrix} = \mathrm{tr}(A)\,B$$

$$\mathrm{tr}_A\left(\begin{bmatrix} [\;\;] & [\;\;] & \cdots & \\ [\;\;] & [\;\;] & & \\ \vdots & & \ddots & \\ & & & [\;\;] \end{bmatrix}\right)$$

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$