Data Structures and Algorithms

Mohammad GANJTABESH

Department of Computer Science,
School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.

mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

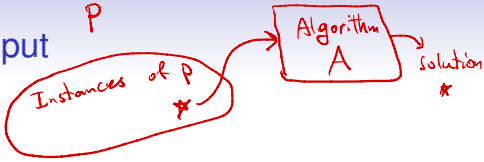
Analysis of Algorithms

Data Structures and Algorithms
Undergraduate course

Mohammad GANJTABESH
mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Size of input



Different input requires different amount of resources.

Primality :

n = 53 ⟶ yes

n = 571426352 ⟶ No

size = 4 , {1000 , ---- 9999}

Size of input

Different input requires different amount of resources.

How about primality test?

Analysis of Algorithms

Predicting/Estimating the amount of resources that the algorithm requires.

Expectation : the amount of resources growth with respect to the size.

size of instance : the amount of memory required for representing the instance (in a fixed coding)

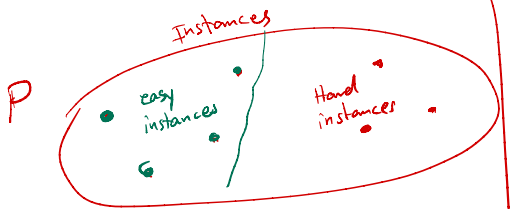
Binary

* For Array : number of elements.
* For Integer : number of digits / bits.
* For Graph : V & E vertices / Edges. or both.

Time complexity : measure the required time.

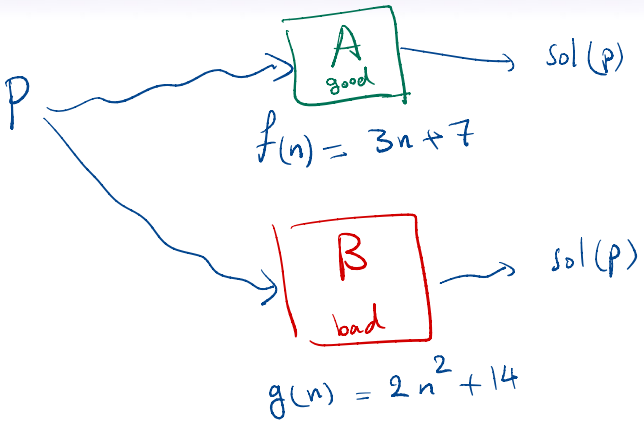
Space ~ : ~ ~ ~ memory

1) Best case : measuring the required amount of resources for easy instances.
2) Worse case : ~ ~ ? = for Hard instances.
3) Average case : ~ the average required amount of resources.



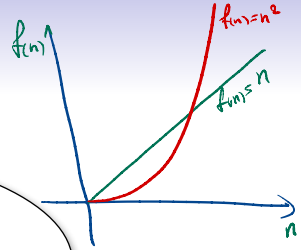
Should be independent from

- the current tech. (Hardware, Prog. Lang. ...)
- the way of implementation.

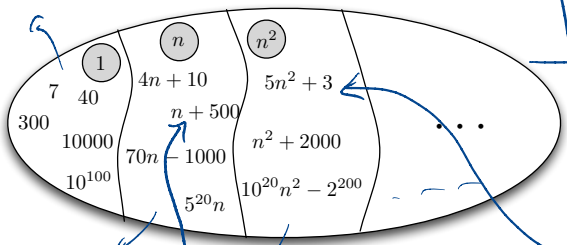

P
A
good
sol(p)
f(n) = 3n + 7
B
bad
sol(p)
g(n) = 2n^2 + 14

$1 \prec \log n \prec n \preceq n \log n \prec n^2 \prec \cdots$

Classifying the functions



constant



Linear

Quadratic

$4n+10 = \Theta(n)$

$c_1 = 5$, $5n$

$c_2 = 3$ $3n$

$n_0 = 3$

A $\Rightarrow$ $f(n) = n+500$

B $\Rightarrow$ $g(n) = 5n^2+3$

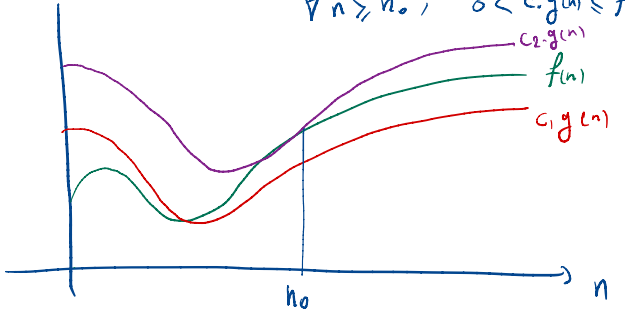
Θ Notation

Upper and Lower bound at the same time.

Let $f, g : \mathbb{N}^* \longrightarrow \mathbb{R}$

$f(n) = \Theta(g(n)) \iff \exists c_1, c_2 > 0, \exists n_0 > 0, \ni$

$\forall n \geq n_0, \quad 0 < c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$



O Notation

Smallest upper bound.

Let $f, g : \mathbb{N}^+ \longrightarrow \mathbb{R}$.

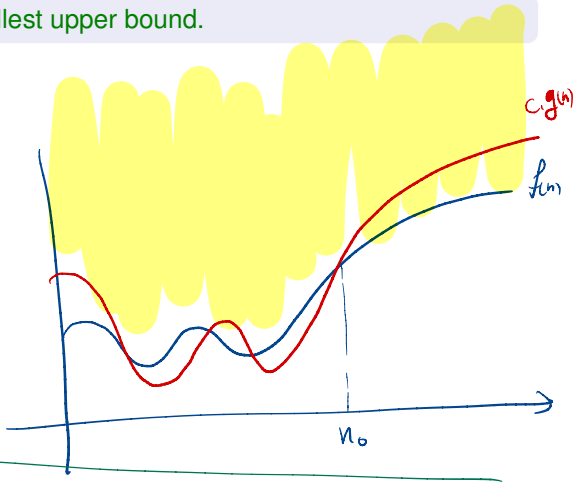
$f(n) = O(g(n))$

$\Longleftrightarrow$

$\exists c > 0, \exists n_0 > 0 \Rightarrow$

$\forall n \geq n_0$

$0 \leq f(n) \leq c \cdot g(n)$



$f(n) = O(g(n)) \Longleftrightarrow \forall c > 0, \exists n_0 > 0 \Rightarrow \forall n \geq n_0, \; 0 \leq f(n) \leq c \cdot g(n)$

Ω Notation

Largest lower bound.

Let $f, g : \mathbb{N}^+ \mapsto \mathbb{R}$

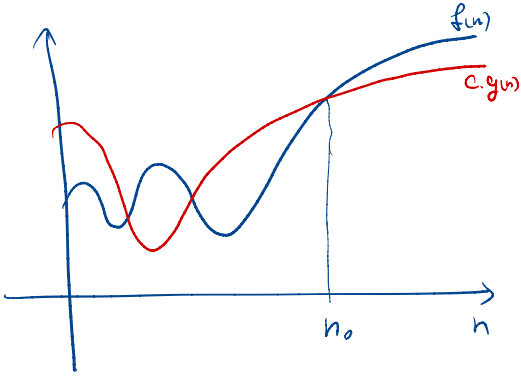
$f(n) = \Omega(g(n))$

$\Leftrightarrow$

$\exists c > 0$, $\exists n_0 > 0$ s.t.

$\forall n \geq n_0$

$0 < c \cdot g(n) \leq f(n)$



$f(n) = \omega(g(n)) \Leftrightarrow \forall c > 0$, $\exists n_0 > 0$ s.t. $\forall n \geq n_0$, $0 < c \cdot g(n) \leq f(n)$

Properties of these notations

Theorem

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)).$$

Properties of these notations

Theorem

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)).$$

Reflexivity

$$f(n) = \Theta(f(n)), f(n) = O(f(n)), \text{ and } f(n) = \Omega(f(n)).$$

Symmetry

$$f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n)).$$

Transitivity

$$\begin{aligned} f(n) = \Theta(g(n)) \text{ and } g(n) = \Theta(h(n)) &\implies f(n) = \Theta(h(n)) \\ f(n) = O(g(n)) \text{ and } g(n) = O(h(n)) &\implies f(n) = O(h(n)) \\ f(n) = \Omega(g(n)) \text{ and } g(n) = \Omega(h(n)) &\implies f(n) = \Omega(h(n)) \end{aligned}$$

Transpose symmetry

$$f(n) = O(g(n)) \iff g(n) = \Omega(f(n)).$$