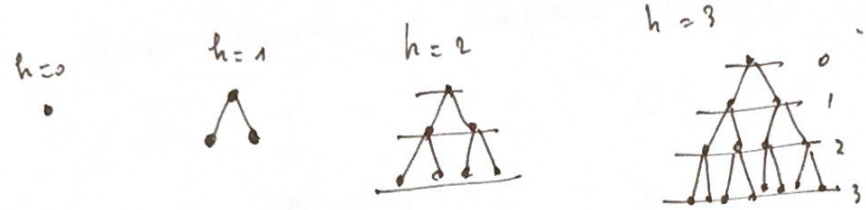
8/01/2020 1/

1) Si consideri un albero binario completo con altezza h ($h=0$ è la radice)

Cosa accade al numero complessivo dei nodi $n(h)$ quando aumenta di un livello.

$$n(h+1) \stackrel{?}{=}$$



(a) $n(h+1) = 2\,n(h)$

(b) $n(h+1) = h\,n(h-1)$

(c) $n(h+1) = 2\,n(h) - 1$

1. $n(h)$ = # nodi sul livello h

$$\boxed{n(h+1) = 2\,n(h)}$$

2 $n(h)$ = # nodi complessivi di un albero di altezza h

$$n(h) = 2^{h+1} - 1 \qquad 1 + 2 + 4 + 8$$

$$1 + 2 + 2^2 + 2^3$$

$$n_h(i) = \qquad 2^{h+1+1} - 2$$

$$n(h+1) = 2^{h+1+1} - 1 = 2\left(2^{h+1} - 1\right) + 1$$

$$= 2\,n(h) + 1$$

$$\boxed{n(h+1) = 2\,n(h) + 1}$$

$n(2) = 2 \cdot 3 + 1$

7

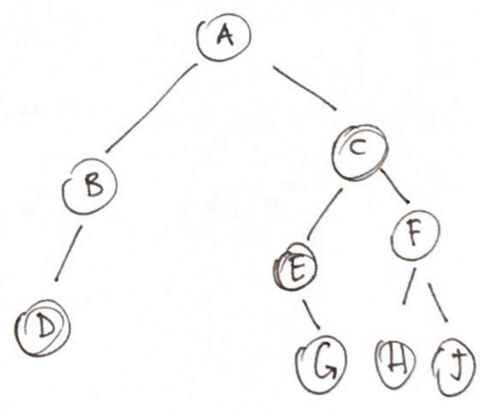
2) Quali delle seguenti affermazioni sono vere a proposito della complessità della visita simmetrica di un albero binario di n nodi?

(a) Ci sono algoritmi di complessità $O(\log n)$

(b) " " " " " $O(\log^2 n)$

(c) " " " " " $\Theta(\log n)$

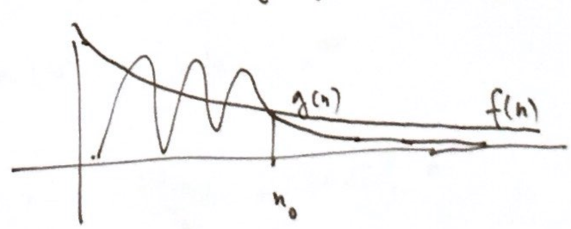
(d) nessuna delle precedenti.



D B A E G C H F J

Cosa vuol dire $O(f(n))$ $f: \mathbb{N} \to \mathbb{N}$?

$g(n) = O(f(n))$ $\left(g(n) \in O(f(n)) \right)$



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

<u>Def.</u> $g(n) = O(f(n))$ se $\exists\, M > 0$ e $n_0 \in \mathbb{N}$ tale che $\forall\, n > n_0$

$$|g(n)| \leq M\,|f(n)|$$

<u>Def.</u> $g(n) = \Omega(f(n))$ se $\exists\, L > 0$ e $n_0 \in \mathbb{N}$ tale che $\forall\, n > n_0$

$$|g(n)| \geq L\,|f(n)|$$

Def $g(n) = \Theta(f(n))$ se $g(n) = O(f(n))$ e $g(n) = \Omega(f(m))$

<u>Ex.</u>
Dimostrare che $P_m(n) = O(n^m)$ dove

$$P_m(n) = a_0 + a_1 n + a_2 n^2 + \dots + a_m n^m$$

$$7n + \cancel{4} n^2 + n^3 = O(n^3)$$

<u>Proof</u>:

[illegible] a

$$|P_m(n)| = |a_0 + a_1 n + a_2 n^2 + \dots + a_m n^m|$$

$$\leq |a_0| + |a_1| n + |a_2| n^2 + \dots + |a_m| n^m$$

$$= \left(\frac{|a_0|}{n^m} + \frac{|a_1|}{n^{m-1}} + \dots \frac{|a_{m-1}|}{n} + |a_m| \right) n^m$$

$$\leq (|a_0| + |a_1| + \dots + |a_m|)\, n^m$$

Basta scegliere

$M = |a_0| + |a_1| + \dots + |a_n|$ $\qquad n_0 = 1$

<u>Ex</u> Data $A \in \mathbb{R}^{n,n}$ $(\mathbb{R}^{n \times n})$

$\mathbb{N}^{n,n}$

Si consideri la classe degli algoritmi che stabiliscono se la somma degli elementi della diagonale principale è il doppio della somma degli elementi della diagonale secondaria.

(a) ~~ke~~ L'algoritmo con massima efficienza ha complessità $O(n^4)$

(b) Esiste un algoritmo con complessità $O(n)$

(c) " " " " " $\Theta(n^3)$

(d) Il problema ~~non~~ è comp. intrattabile.

$n = O(n^4)$

<u>Ex</u> Quanti archi ha un grafo completo con n vertici? K_n

è $\dfrac{n(n-1)}{2}$

<u>Ex</u> Dato un albero binario con 1024 nodi qual è la sua profondità minima?

Basta considerare gli alberi binari completi

nodi $\quad 2^{h+1} - 1 \qquad \boxed{h = 10}$