

ESAME 26 GIUGNO 2012 (SOLO ESERCIZI 1 e 2)

ESERCIZIO 1. (10 punti)

Data la seguente equazione di ricorrenza:

$$T(n) = 16T(n/4) + \Theta(n^2)$$

$$T(1) = \Theta(1)$$

- a. (3 punti) si risolve l'equazione applicando il metodo del teorema principale;
b. (7 punti) si risolve l'equazione applicando il metodo iterativo.

METODO PRINCIPALE

$$n^{\log_b a} = n^{\log_4 16} = n^4$$

$$f(n) = \Theta(n^2)$$

$$f(n) = O(n^{4-\epsilon}) \text{ quindi } T(n) = \Theta(n^4)$$

METODO ITERATIVO

$$T(n) = 16 \left[16T\left(\frac{n}{4^2}\right) + \Theta\left(\left[\frac{n}{4}\right]^2\right) \right] + \Theta(n^2)$$

$$T(n) = 16 \left[16 \left[16T\left(\frac{n}{4^3}\right) + \Theta\left(\left[\frac{n}{4^2}\right]^2\right) \right] + \Theta\left(\left[\frac{n}{4}\right]^2\right) \right] + \Theta(n^2)$$

$$T(n) = 16^K T\left(\frac{n}{4^K}\right) + \sum_{i=0}^{K-1} 16^i \Theta\left(\left[\frac{n}{4^i}\right]^2\right) \text{ dove } 4^K = n \Rightarrow K = \log_4 n$$

$$T(n) = 16^{\log_4 n} \Theta(1) + \sum_{i=0}^{\log_4 n - 1} 16^i \Theta\left(\frac{n^2}{4^{2i}}\right)$$

$$T(n) = n^4 \Theta(1) + \sum_{i=0}^{\log_4 n - 1} \Theta\left(\frac{16^i n^2}{4^{2i}}\right) = \Theta(n^4) + \sum_{i=0}^{\log_4 n - 1} \Theta\left(\frac{\cancel{16}^i n^2}{\cancel{16}^i}\right)$$

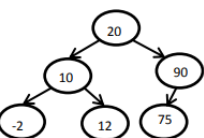
$$= \Theta(n^4) + \sum_{i=0}^{\log_4 n - 1} \Theta(n^2) = \Theta(n^4) + \Theta(\log_4 n \cdot n^2) = \Theta(n^4)$$

ESERCIZIO 2. (10 punti)

Progettare una funzione che, presi:

- un vettore A di n numeri interi;
 - un albero binario di ricerca (ABR) implementato tramite puntatori, costituito di m nodi e contenente un numero intero nel campo key di ciascun nodo;
- restituisca 1 se ogni valore contenuto nel vettore è presente nell'ABR, 0 altrimenti.

Ad esempio, consideriamo $n = 3$ ed il seguente albero di $m = 6$ nodi:



Se $A = [20, 75, -2]$, la funzione deve restituire 1
 Se $A = [20, 75, 4]$, la funzione deve restituire 0

Dell'algoritmo progettato:

- (2 punti) si dia la descrizione a parole;
- (6 punti) si dia lo pseudocodice, specificando il valore dei parametri alla prima chiamata;
- (2 punti) si valuti la complessità della soluzione proposta, sotto l'ipotesi che l'ABR sia completo o quasi completo.

COME PRIMA COSA TROVERO' IL MASSIMO DELL'ABR, E DEFINIRO' UNA VARIABILE "GAP", DATA DAL MASSIMO + 1, DOPO DI CHE VISITERO' OGNI NODO, CONTROLLANDO SE LA CHIAVE E' PRESENTE NEL VETTORE, SE SI, SOSTITUIRO' QUEL VALORE CON "GAP", INFINE, CONTROLLERO' IL VETTORE, SE UN SOLO VALORE SARA' DIVERSO DA GAP, RESTITUIRO' 0, ALTRIMENTI 1;

DEF FIND_GAP(ABR): $T(n) = O(n)$

IF (!ABR->RIGHT):

RETURN KEY+1;

ELSE: RETURN FIND_GAP(ABR->RIGHT);

DEF CONTROLLO_ABR(ABR, A, GAP): $T(n, m) = \Theta(m \cdot n)$

IF (!ABR): RETURN

FOR i IN RANGE(LEN(A)-1): n VOLTE

IF (A[i] == ABR->KEY):

A[i] = GAP;

CONTROLLO_ABR(ABR->LEFT, A, GAP); $T(k)$

CONTROLLO_ABR(ABR->RIGHT, A, GAP); $T(m-k-1)$

DEF ES2(ABR, A): MAIN FUNCTION

GAP = FIND_GAP(ABR); $O(m)$

CONTROLLO_ABR(ABR, A, GAP); $\Theta(m \cdot n)$

FOR i IN RANGE(LEN(A)-1): $\Theta(n)$

IF (A[i] != GAP):

RETURN 0;

RETURN 1;

$T(n) = \Theta(m \cdot n)$