

Algoritmi e Strutture Dati 1

Appello del 22/12/2010

Esercizio 1

Scrivere un metodo `public static boolean LEQ (int [] A, int[] B)` in Java che presi in input due array **A** e **B** di n ed m oggetti rispettivamente, restituisce *true* se e solo se ogni elemento di A è minore o uguale di tutti gli elementi di B.

La complessità computazionale della procedura deve essere $\theta(n + m)$. Dimostrare formalmente il risultato di complessità del metodo proposto.

Esercizio 2

$$\text{Sia } T(n) = \begin{cases} T(n/3) + c_1 & \text{se } n > 1 \\ c_2 & \text{se } n = 1 \end{cases}$$

dove c_1 e c_2 sono due costanti.

Si dia una stima esplicita (non ricorsiva) di $T(n)$ facendo uso

- del **metodo di sostituzione ed induzione**;
- del **teorema generale** (*mostrare formalmente perché è valido il caso che si applica*).

Esercizio 3

- a. **Mostrare** l'albero binario di ricerca che si ottiene a partire dall'albero vuoto inserendo nell'ordine indicato i seguenti elementi:

13, 2, 1, 18, 60, 24, 59, 100, 3, 34

Che altezza ha l'albero ottenuto?

- b. Dopo aver scritto la classe **Node** vista a lezione, scrivere un metodo ricorsivo `public static int SUM (Node x)` che preso in input un nodo **x**, restituisce la somma delle chiavi contenute nell'albero radicato al nodo **x**. Si discuta la complessità computazionale della procedura proposta.

Esercizio 4

Si consideri una *tabella hash con lista di trabocco* contenente numeri interi. Data la funzione hash **modulo**, si mostri un esempio in cui l'inserimento di **n chiavi intere** $k_1, k_2, k_3, \dots, k_n$, in una tabella con **31** posizioni risulti in una tabella in cui la ricerca di una chiave **k** risulta avere il peggior tempo possibile di esecuzione. In particolare, si fornisca per ogni i da 1 a n il valore di k_i (in funzione dell'indice i), ed il valore **k** della chiave da cercare.

Attenzione:

- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso **per nessun motivo** l'uso di telefoni cellulari, calcolatrici, etc...
- **non** è possibile consultare appunti, libri, dispense.