

INFORMATICA GENERALE

Compito Scritto

docenti: TIZIANA CALAMONERI, IVANO SALVO
Sapienza Università di Roma

24 settembre 2012

Esercizio 1 (10 punti) Usando uno dei metodi noti, a scelta, si risolva la seguente equazione di ricorrenza (**4 punti**):

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

$$T(n) = T(n/2) + \Theta(n^2 \log n) \text{ per } n > 1$$

Si risponda poi ad ognuna delle seguenti domande, motivando la risposta:

1. (**2 punti**) Se un algoritmo ha complessità $\Theta(n^2)$ nel caso peggiore, posso dedurre che nel caso migliore terminerà in $\Theta(n^2)$ passi?
2. (**2 punti**) Se si dimostra che un algoritmo ha complessità $\Omega(n^2)$ nel caso migliore, è possibile che in qualche caso l'algoritmo termini in $O(n)$ passi?
3. (**2 punti**) Se si dimostra che un algoritmo ha complessità $\Theta(n^2)$ nel caso peggiore, è possibile che in qualche caso l'algoritmo termini in $O(n)$ passi?

Esercizio 2 (12 punti) Si scriva una funzione C `lista mergeSort(lista L)` che ordini la lista di interi `L`, adattando opportunamente l'algoritmo `mergeSort` alle liste. Procedere come segue:

1. Considerare come caso base la lista vuota;
2. Scrivere una funzione `dividi` che presa una lista con almeno 1 elemento, in un'unica scansione sposti i puntatori in modo tale che generi due liste, una contenente gli elementi in posizione pari, l'altra contenente gli elementi in posizione dispari.

3. chiamare ricorsivamente la funzione `mergeSort` sulle due liste;
4. scrivere una funzione `merge` per fondere ordinatamente le liste ordinate ottenute dalle chiamate ricorsive della funzione `mergeSort` modificando opportunamente i puntatori.

Discutere brevemente vantaggi e svantaggi dell'algoritmo `mergeSort` sulle liste rispetto alla versione su array e fare le opportune osservazioni sulla complessità.

N.B. Il codice deve essere opportunamente commentato.

Esercizio 3 (10 punti) Dato un grafo non orientato $G(V, E)$ memorizzato tramite liste di adiacenza, progettare un algoritmo che dia in output 1 se G aciclico e 0 in caso contrario. Dell'algoritmo proposto:

1. (**2 punti**) si dia la descrizione a parole;
2. (**3 punti**) si scriva lo pseudocodice;
3. (**2 punti**) si valuti la complessità;
4. (**3 punti**) si facciano le opportune considerazioni sulla complessità nel caso in cui il grafo sia memorizzato tramite matrice di adiacenza e matrice di incidenza.