

INFORMATICA GENERALE

Esame Scritto

docenti: G. BONGIOVANNI, T. CALAMONERI, I. SALVO
Sapienza Università di Roma

21 settembre 2016

Esercizio 1 (10 punti) Si consideri la seguente equazione di ricorrenza:

$$T(n) = T(n - 1) + \Theta(n)$$
$$T(1) = \Theta(1).$$

Si risolva tale equazione di ricorrenza utilizzando:

- (2 punti) il metodo iterativo;
- (2 punti) il metodo dell'albero;
- (3 punto) il metodo di sostituzione;

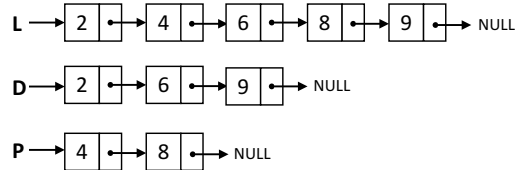
Descrivere brevemente un problema affrontato a lezione che possa essere risolto con un algoritmo il cui costo computazione sia espresso dall'equazione data e scrivere lo pseudocodice dell'algoritmo studiato (3 punti).

Esercizio 2 (10 punti) Considerare il problema di dividere una lista di interi in due liste, una contenente tutti gli elementi che occorrono in posizione pari e una contenente tutti gli elementi in posizione dispari. Scrivere due funzioni in linguaggio C:

- (5 punti) La prima, void divideFun(lista L, lista* D, lista *P), *alloca* nuova memoria e copia gli elementi di posto dispari nella lista D e gli elementi di posto pari nella lista P.
- (5 punti) La seconda void divide(lista L, lista* D, lista *P) modifica la lista originaria L modificando adeguatamente i puntatori restituendo nella variabile P un puntatore alla testa della lista degli elementi di posto pari e nella variabile D un puntatore alla testa della lista degli elementi di posto dispari.

ESEMPIO: Entrambe le funzioni, se la lista di ingresso L fosse $\langle 2, 4, 6, 8, 9 \rangle$, dovrebbero costruire le liste $D = \langle 2, 6, 9 \rangle$ e $P = \langle 4, 8 \rangle$. Tuttavia, i loro effetti in memoria saranno molto diversi, come esemplificato in Fig. 1.

a) risultato in memoria dopo l'esecuzione di `divideFun(L, &D, &P)`



b) risultato in memoria dopo l'esecuzione di `divide(L, &D, &P)`

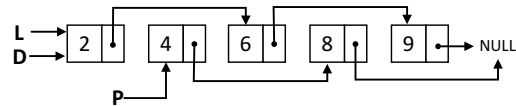


Figura 1: Effetti in memoria delle due funzioni.

Esercizio 3 (10 punti) Siano dati un albero binario di ricerca T memorizzato tramite records e puntatori ed un valore intero a .

Progettare una funzione il più efficiente possibile che stampi tutte le chiavi di T che risultano maggiori di a .

Della funzione proposta:

1. **(3 punti)** Si dia la descrizione a parole;
2. **(3 punti)** Si scriva lo pseudocodice e si mostri la chiamata iniziale;
3. **(2 punti)** Si calcoli il costo computazionale (che dovrebbe essere $O(h + k)$, dove h è l'altezza di T e k è il numero di chiavi stampate);
4. **(2 punti)** Si mostri il funzionamento della funzione su due esempi diversi in cui il numero di nodi dell'albero sia lo stesso, ma il costo computazionale sia differente.