

Introduzione agli Algoritmi

Esame Scritto a canali unificati

docenti: T. Calamoneri, A. Monti
Sapienza Università di Roma
24 Ottobre 2024

Esercizio 1 (10 punti): Siano date le tre seguenti equazioni di ricorrenza:

$$\begin{array}{lll} T(n) = 4T(\frac{n}{2}) + \Theta(n) & T(n) = 4T(\frac{n}{2}) + \Theta(n^2) & T(n) = 4T(\frac{n}{2}) + \Theta(n^3) \\ T(n) = \Theta(1) & T(n) = \Theta(1) & T(n) = \Theta(1) \end{array}$$

Si risolvano con il teorema principale, dettagliando il ragionamento usato ed evidenziando le differenze.

Esercizio 2 (10 punti): Si scriva lo pseudocodice, opportunamente commentato, di una funzione **iterativa** che, preso in input un array A di interi, trovi la lunghezza massima delle sequenze crescenti presenti nell'array.

Ad esempio:

- per $A = [3, 1, 5, 2, 6, 8, 7, 1]$ la risposta è 3 che è la lunghezza della sequenza 2, 6, 8
- per $A = [10, 7, 6, 1]$ la risposta è 1 perché non ci sono in A sequenze crescenti di lunghezza maggiore ad 1

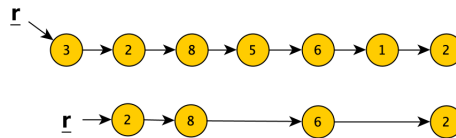
La funzione deve avere costo computazionale $\mathcal{O}(n)$, dove n è il numero di elementi presenti nell'array. Il costo in termini di spazio oltre l'array A deve essere $\Theta(1)$ (in pratica non si può far uso di array di appoggio).

Il costo computazionale dell'algoritmo proposto va dettagliato valutando il contributo di ogni istruzione.

Esercizio 3 (10 punti): Si consideri una lista concatenata, i cui nodi abbiano un campo *key* contenente un intero ed un campo *next* al nodo successivo (next vale *None* se si tratta dell'ultimo nodo).

Si progetti una funzione **ricorsiva** che, dato il puntatore *r* alla testa della lista restituisca il puntatore alla testa della lista modificata in modo che vi compaiano solo i nodi contenenti gli interi pari.

Ad esempio, data la lista concatenata *r* della figura, la funzione deve restituire il puntatore alla nuova lista concatenata contenente i soli nodi con valore pari.



La funzione deve avere costo computazionale $\mathcal{O}(n)$ dove *n* è il numero di nodi presenti nella lista originaria e l'algoritmo non può generare nuovi nodi nè utilizzare array di appoggio. Della funzione proposta:

- a) si dia la descrizione a parole,
- b) si scriva lo pseudocodice,
- c) si giustifichi formalmente il costo computazionale dando la ricorrenza che lo caratterizza e poi la sua soluzione.

NOTA BENE: nello pseudocodice della funzione ricorsiva **non** si deve far uso di variabili globali.