

Esame di algoritmi e strutture dati

13 maggio 2020

Tempo a disposizione: 2 ore

Esercizio 1

(7 punti)

1. Progettare un algoritmo (pseudocodice):

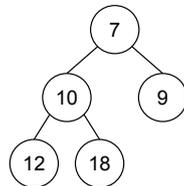
$quasiHeap(AlberoBinario T) \rightarrow Booleano$

che, preso in input un albero binario T contenente valori interi come chiavi, restituisca *true* se e solo se ciascun nodo N dell'albero contiene una chiave con valore maggiore o uguale rispetto al valore delle chiavi dei nodi contenuti in entrambi i sottoalberi di N .

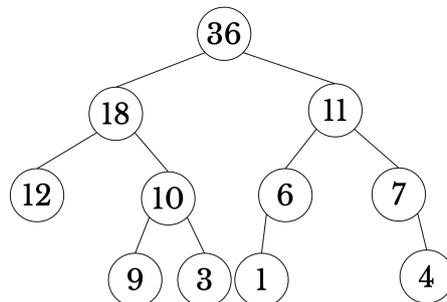
2. Determinare il costo temporale dell'algoritmo.

ESEMPIO.

Con il seguente albero di input, l'algoritmo deve restituire il valore *false*.



Con il seguente albero di input, l'algoritmo deve restituire il valore *true*.



Esercizio 2

(7 punti)

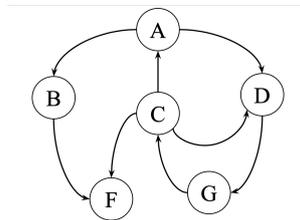
1. Scegliere un algoritmo di ordinamento che abbia costo temporale $\mathcal{O}(n^2)$ e mostrarne lo pseudocodice.
2. Mostrare i passi eseguiti dall'algoritmo scelto quando viene ordinato il seguente array di input:

8	1	12	7	3
---	---	----	---	---
3. Motivare il costo dell'algoritmo.

Esercizio 3

(7 punti)

1. Mostrare lo pseudocodice dell'algoritmo di visita in ampiezza di un grafo orientato.
2. Indicare il costo dell'algoritmo.
3. Mostrare l'ordine di visita dei nodi nel grafo seguente, iniziando la visita dal nodo A :



Esercizio 4

(6 punti)

1. Fornire la definizione di albero binario di ricerca.
2. Mostrare un esempio di albero binario di ricerca avente profondità 4 e contenente 10 nodi.
3. Mostrare la procedura di ricerca nell'albero mostrato, quando viene cercata una chiave non presente nell'albero.
4. Illustrare il costo dell'operazione di ricerca.

Esercizio 5

(6 punti)

Risolvere la seguente equazione di ricorrenza usando il metodo che si ritiene più appropriato:

- $T(n) = 1$, se $n \leq 1$
- $T(n) = 8T(\frac{n}{2}) + 4n^3 + 2n^2$, altrimenti