

Algoritmi e Strutture Dati

a.a. 2022/23

Compito del 5/6/2023

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

Avvertenza: Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia H un vettore i cui elementi costituiscono un **Max-heap** contenente n interi. Si indichi quali tra le seguenti affermazioni sono corrette, **motivando brevemente le risposte**:
 - a. la ricerca di un intero k in H ha costo $O(\log n)$;
 - b. H può contenere chiavi ripetute;
 - c. la chiave minima si trova sicuramente in una foglia;
 - d. la chiave minima si trova sicuramente in $H[\text{heapsize}[H]]$.
2. Un algoritmo ricorsivo \mathcal{A} accetta in ingresso una costante k e un grafo G connesso non orientato e pesato, e restituisce TRUE se esiste in G un albero di copertura di peso minore o uguale a k e FALSE in caso contrario. La sua complessità è data da

$$T(m) = 16T\left(\frac{m}{4}\right) + 5m + 7$$

dove m rappresenta il numero di archi di G . Esistono algoritmi più efficienti di \mathcal{A} per risolvere il problema dato?

3. Sia dato un grafo non orientato connesso $G = (V, E, w)$ con pesi positivi e si supponga di eseguire l'algoritmo di Kruskal su G sostituendo i pesi originali come segue:
 - a) $w(u,v) \rightarrow 2 + w(u,v)$
 - b) $w(u,v) \rightarrow 2w(u,v)$
 - c) $w(u,v) \rightarrow 1 / w(u,v)$
 - d) $w(u,v) \rightarrow \log w(u,v)$
 - e) $w(u,v) \rightarrow w(u,v)^2$

In quali casi si otterranno gli stessi risultati del grafo originale? Perché?

Algoritmi e Strutture Dati

a.a. 2022/23

Compito del 5/6/2023

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

Avvertenza: Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia T un BST di dimensione n e sia k una chiave di T tale che $T \rightarrow \text{root} \rightarrow \text{key} < k$. Si vuole costruire un **nuovo** BST T' contenente tutte e sole le chiavi di T appartenenti all'intervallo $[T \rightarrow \text{root} \rightarrow \text{key}, k]$.
 - a. Si scriva una funzione **efficiente** in C o C++ per risolvere il problema. Il prototipo della funzione è:
`PTree creaBSTInterval(PTree t, int k)`
 - b. Valutare e giustificare la complessità della funzione.
 - c. Specificare il linguaggio di programmazione scelto e la definizione di `PTree`.

2. Scrivere un algoritmo che date due sequenze X e Y , rispettivamente di m e n caratteri, restituisca la lunghezza di una sottosequenza di lunghezza massima comune a X e Y (LCS). Più precisamente:
 - a. dare una caratterizzazione ricorsiva della lunghezza di una LCS, definendo $\text{lunq}[i, j]$ come la lunghezza di una LCS delle sequenze X^i e Y^j con $0 \leq i \leq m$ e $0 \leq j \leq n$. Si ricordi che X^i è il prefisso di X di lunghezza i e Y^j è il prefisso di Y di lunghezza j ;
 - b. tradurre tale definizione in un algoritmo di programmazione dinamica con il metodo **top-down** che determina la lunghezza di una LCS;
 - c. valutare e giustificare la complessità dell'algoritmo.

3. Il **diametro** di un grafo pesato è definito come la *massima distanza tra due vertici* del grafo, dove, come è noto, per "distanza tra due vertici" u e v si intende la lunghezza del cammino minimo tra u e v . Si sviluppi un algoritmo efficiente (ovvero polinomiale) che accetti in ingresso un grafo G e ne calcoli il suo diametro (si assuma che G non contenga cicli negativi). Si dimostri la correttezza dell'algoritmo proposto e si determini la sua complessità computazionale.

Nota. Nel caso in cui si faccia uso di algoritmi noti si dimostri anche la loro correttezza.

4. Si scriva l'algoritmo di Dijkstra, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo rappresentato mediante lista di adiacenza (utilizzando il vertice 0 come sorgente):

[illegible]