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)
	Si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli erranno valutati pienamente.
	Tun vettore i cui elementi costituiscono un Max-heap contenente <i>n</i> interi. Si indichi quali tra le seguenti mazioni sono corrette, motivando brevemente le risposte : a. la ricerca di un intero <i>k</i> in <i>H</i> ha costo $O(\log n)$; b. <i>H</i> può contenere chiavi ripetute; c. la chiave minima si trova sicuramente in una foglia; d. la chiave minima si trova sicuramente in $H[heapsize[H]]$.
restiti	goritmo ricorsivo \mathcal{A} accetta in ingresso una costante k e un grafo G connesso non orientato e pesato, e aisce TRUE se esiste in G un albero di copertura di peso minore o uguale a k e FALSE in caso contrario a 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 G

- 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

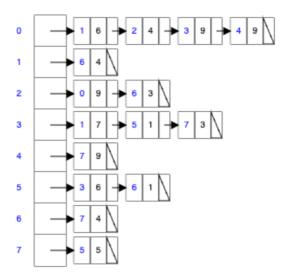
a.a. 2022/23

Compito del 5/6/2023

Cognome:	Nome:
Matricola:	E-mail:
	Parte II (2.5 ore; ogni esercizio vale 6 punti)
Avvertenza: Si giustifichino tecnicar esercizi non verranno valutati piename	mente tutte le risposte. In caso di discussioni poco formali o approssimative gli ente.
BST T' contenente tutte e sole a. Si scriva una funz PTree creaBS b. Valutare e giustifi	m e sia k una chiave di T tale che $T \rightarrow root \rightarrow key < k$. Si vuole costruire un nuovo e le chiavi di T appartenenti all'intervallo $[T \rightarrow root \rightarrow key, k]$. ione efficiente in C o $C++$ per risolvere il problema. Il prototipo della funzione è: TInterval (PTree t, int k) care la complessità della funzione. uaggio di programmazione scelto e la definizione di PTree.
una sottosequenza di lunghez a. dare una caratterizza di una LCS delle se lunghezza i e Y è il j b. tradurre tale definiz determina la lunghez	te due sequenze X e Y , rispettivamente di m e n caratteri, restituisca la lunghezza di za massima comune a X e Y (LCS). Più precisamente: zione ricorsiva della lunghezza di una LCS, definendo $lung[i,j]$ come la lunghezza equenze X^i e Y^i con $0 \le i \le m$ e $0 \le j \le n$. Si ricordi che X^i è il prefisso di X di prefisso di Y di lunghezza j ; ione in un algoritmo di programmazione dinamica con il metodo top-down che zza di una LCS; te la complessità dell'algoritmo.
per "distanza tra due vertici" tefficiente (ovvero polinomial	o è definito come la massima distanza tra due vertici del grafo, dove, come è noto, $u \in v$ si intende la lunghezza del cammino minimo tra $u \in v$. Si sviluppi un algoritmo e) che accetti in ingresso un grafo G e ne calcoli il suo diametro (si assuma che G Si dimostri la correttezza dell'algoritmo proposto e si determini la sua complessità

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):



In particolare:

- a) si indichi l'ordine con cui vengono estratti i vertici b) si riempia la tabella seguente con i valori dei vettori de π , iterazione per iterazione:

	vertice 0		vertice 1		vertice 2		vertice 3		vertice 4		vertice 5		vertice 6		vertice 7	
	d	π	d	π	d	π	d	π	d	π	d	π	d	π	d	π
dopo INIT_SS																
iter 1																
iter 2																
iter 3																
iter 4																
iter 5																
iter 6																
iter 7																
iter 8																