

Algoritmi e Strutture Dati

a.a. 2024/25

Compito del 04/06/2025

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. Si disegni l'albero binario di ricerca la cui visita in post-ordine ha come risultato 10, 21, 41, 39, 45, 47, 43, 35. Poi si effettui la cancellazione del nodo 35 con la procedura *Tree-Delete* e si disegni l'albero risultante.

2. Un algoritmo per determinare se in un grafo pesato esistano cicli negativi ha complessità

$$T(n) = 8T(n/2) + n^3$$

dove n rappresenta il numero di nodi del grafo in ingresso. Si dica se, o in quali casi, l'algoritmo in questione è preferibile all'algoritmo *Iterated BF*, che consiste nell'eseguire n volte l'algoritmo di Bellman-Ford (ciascuna iterazione con un vertice sorgente diverso).

3. Un algoritmo per determinare se in grafo non orientato esista un ciclo che passi per tutti i vertici (ciclo Hamiltoniano) ha complessità

$$T(n) = 3T(n/2) + n^2$$

dove n rappresenta il numero di nodi del grafo in ingresso. È plausibile che l'algoritmo in questione sia corretto? Perché?

Algoritmi e Strutture Dati

a.a. 2024/25

Compito del 04/06/2025

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 albero generale i cui nodi contengono interi e hanno campi: **key**, **left-child** e **right-sib**. Un nodo u di T è detto **intermedio** se la somma delle chiavi contenute nei nodi del sottoalbero di cui u è radice è uguale alla somma delle chiavi contenute nei nodi sul percorso che collega u alla radice dell'albero (u escluso).
 - a. Si scriva una funzione **efficiente** in C o C++ che restituisca il numero di nodi intermedi.
 - b. Valutare e giustificare la complessità della funzione.
 - c. Specificare il linguaggio di programmazione scelto e la definizione di `PNodeG`.

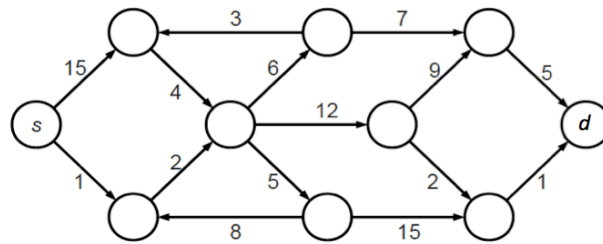
Il prototipo della funzione è

```
int intermedi(PNodeG u)
```

2. Si consideri la struttura dati Max-Heap implementata tramite un vettore (secondo lo schema visto a lezione).
 - a) Si scriva la definizione di Max-Heap.
 - b) Si realizzi in modo **efficiente** la funzione `Intersezione(H1, H2)` che dati due Max-Heap $H1$ e $H2$ contenenti rispettivamente $n1$ e $n2$ interi (anche ripetuti), ritorna in output un **nuovo** Max-Heap contenente gli elementi che appartengono sia a $H1$ che a $H2$. In presenza di duplicati se x compare $k1$ volte in $H1$ e $k2$ volte in $H2$, nel Max-Heap intersezione x dovrà comparire $\min\{k1, k2\}$.
 - c) Si determini e giustifichi la complessità in funzione di $n1$ e $n2$.

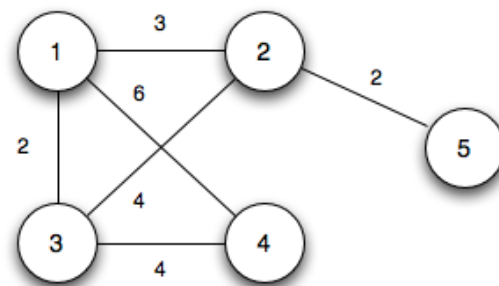
Si devono scrivere le eventuali funzioni/procedure ausiliarie utilizzate.

3. La rete ferroviaria italiana può essere descritta mediante un grafo orientato pesato $G = (V, E, w)$, dove i vertici rappresentano le stazioni, la presenza di un arco tra due vertici indica l'esistenza di una tratta ferroviaria diretta tra le corrispondenti stazioni e, per ogni arco $(u, v) \in E$, il peso $w(u, v)$ rappresenta la quantità di carburante necessaria per raggiungere la stazione v partendo da u . Si scriva un algoritmo che, dati in ingresso il grafo G , la quantità C di carburante inizialmente presente nel serbatoio della locomotiva di un treno, e due nodi s e d , restituisca **TRUE** se esiste un cammino che consente al treno di raggiungere la stazione d partendo dalla stazione s , e **FALSE** in caso contrario. Si discuta della correttezza e della complessità computazionale dell'algoritmo proposto e si simuli accuratamente la sua esecuzione sul seguente grafo, con $C = 18$.



Nota: nel caso in cui si utilizzi un algoritmo noto, si scriva esplicitamente l'algoritmo e se ne dimostri la correttezza.

4. Si scriva l'algoritmo di Prim, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo (utilizzando il vertice 1 come “sorgente”):



In particolare:

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

[illegible]