

Algoritmi e Strutture Dati

a.a. 2018/19

Compito del 12/09/2019

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

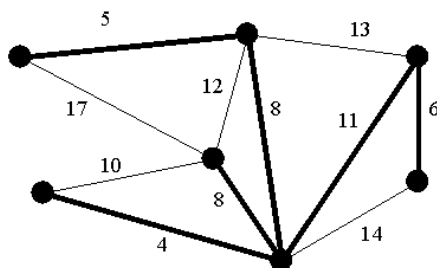
1. Scrivere l'algoritmo build-Max-Heap e simulare la sua esecuzione sull'array $\langle -10, -3, -7, 15, 12, 36 \rangle$

2. Il Prof. H. A. Milton sostiene di aver sviluppato un algoritmo di complessità

$$T(n) = 9T\left(\frac{n}{3}\right) + n^2$$

che riceve in ingresso un grafo non orientato G con n vertici e risponde TRUE se esiste in G un ciclo che passa per tutti i suoi vertici, e FALSE in caso contrario. Si dica, **giustificando tecnicamente la risposta**, se l'affermazione è verosimile.

3. Si dica, **giustificando tecnicamente la risposta**, se nel grafo sottostante gli archi indicati in grassetto formano o meno un albero di copertura minimo.



Algoritmi e Strutture Dati

a.a. 2018/19

Compito del 12/09/2019

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

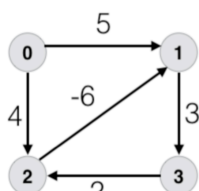
Parte II

(2.5 ore; ogni esercizio vale 6 punti)

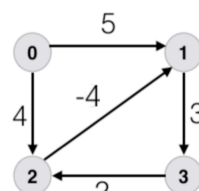
1. Sia T un albero binario di ricerca. Si vogliono stampare le chiavi di T memorizzate in nodi il cui sottoalbero radicato nel figlio sinistro contiene un numero pari di chiavi e il sottoalbero radicato nel figlio destro contiene un numero dispari di chiavi.
 - a. Si rappresenti un albero binario di ricerca la cui visita in pre-order ha come risultato 10, 5, 1, 20, 15, 25. Si mostri quali chiavi verrebbero stampate in base alla condizione sopra descritta.
 - b. Scrivere una **procedura in C efficiente**, di nome `stampaNodi(u)`, che data la radice di un albero binario di ricerca, stampa le chiavi dei nodi che soddisfano la condizione specificata. Valutarne la complessità, **indicando eventuali relazioni di ricorrenza**.
2. Siano dati in input k vettori A_1, \dots, A_k di numeri naturali, ognuno ordinato in modo decrescente. Sia n la quantità di elementi presenti complessivamente nei vettori, ovvero $n = \sum_{i=1}^k A_i.length$. Si consideri il problema di produrre in output il vettore ordinato in modo decrescente B , unione con ripetizioni di A_1, \dots, A_k .
 - a. Scrivere tramite pseudo-codice una procedura **efficiente** per risolvere il problema proposto nel caso in cui k sia costante rispetto ad n . Si determini la complessità.
 - b. Scrivere lo pseudo-codice di una procedura per risolvere il problema proposto nel caso generico avente complessità $O(n \log k)$. Si determini la complessità.
3. Si stabilisca quale problema risolve il seguente algoritmo, che accetta in ingresso un grafo orientato e pesato $G = (V, E)$, la sua funzione peso $w : E \rightarrow \mathbb{R}$, e un vertice $s \in V$:

```
MyAlgorithm(G, w, s)
1.  n = |V[G]|
2.  for each u ∈ V[G] \ {s}
3.    d[u] = +∞
4.  d[s] = 0
5.  for i = 1 to n - 1
6.    for each (u,v) ∈ E[G] do
7.      d[v] = min { d[v], d[u] + w(u,v) }
8.  for each (u,v) ∈ E[G] do
9.    if d[v] != min { d[v], d[u] + w(u,v) } then
10.     print("Yes")
11.   otherwise
12.     print("No")
13. return
```

Si dimostri la correttezza dell'algoritmo e si determini la sua complessità computazionale. Cosa restituisce l'algoritmo in presenza dei seguenti grafi, ponendo $s = 0$?



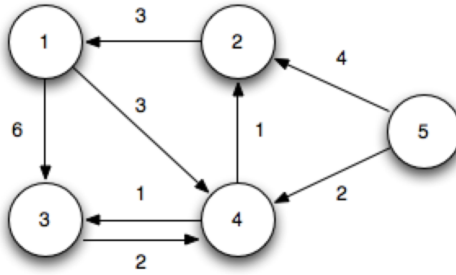
(a)



(b)

Nota: si fornisca una dimostra "completa" della correttezza dell'algoritmo e si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative l'esercizio non verrà valutato pienamente.

4. Si scriva l'algoritmo di Floyd-Warshall, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo:



Nota: si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative l'esercizio non verrà valutato pienamente.