Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 18 settembre 2020

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1 e 2 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 3 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su TUTTI i fogli (inclusi quelli di brutta).

1. Considerate l'albero AVL che si ottiene inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da un albero AVL inizialmente vuoto:

`14 28 33 16 19 22 23 20`

(a) Disegnate l'albero ottenuto

(b) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ampiezza

(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in profondità in ordine anticipato

(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in profondità in ordine simmetrico

(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in profondità in ordine posticipato

(f) L'albero ottenuto al punto (a) è un albero perfettamente bilanciato?

(g) Se al punto (f) avete risposto SI scrivete la definizione di albero perfettamente bilanciato.
Se avete risposto NO indicate il contenuto di tutti i nodi che non rispettano la condizione di bilanciamento relativa agli alberi perfettamente bilanciati.

(h) Disegnate l'albero 2-3 che si ottiene inserendo il un albero 2-3 inizialmente vuoto la sequenza di numeri precedente, nell'ordine indicato.

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:
`14 33 28 16 25 22 23 20`

(a) Supponete di ordinare la sequenza mediante l'algoritmo `radixSort` (rispetto alla base 10). Indicate il contenuto dell'array dopo la prima iterazione del ciclo principale dell'algoritmo.
(b) Supponete di ordinare la sequenza mediante l'algoritmo `selectionSort`. Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.
(c) Supponete di ordinare la sequenza mediante l'algoritmo `bubbleSort`. Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.
(d) Supponete di ordinare la sequenza mediante l'algoritmo `heapSort`. Indicate il contenuto dell'array immediatamente dopo avere effettuato lo scambio che colloca il numero `28` nella sua posizione finale all'interno dell'array.

3. Per definire le tariffe dei mezzi pubblici, gli amministratori di una grande città hanno diviso il territorio in tre fasce, individuate in base alla distanza dal centro. Come *distanza di un luogo dal centro*, si intende la lunghezza del più breve percorso che permette di raggiungere il luogo partendo dal centro.

Le tre fasce sono definite come segue, dove r indica la distanza del luogo della città più lontano dal centro:

- la *fascia centrale* è costituita dai luoghi che si trovano a distanza $d \leq \frac{1}{3}r$ dal centro,
- la *fascia semicentrale* è costituita dai luoghi che si trovano a distanza d con $\frac{1}{3}r < d \leq \frac{2}{3}r$ dal centro,
- la *fascia periferica* è costituita dai luoghi che si trovano a distanza $d > \frac{2}{3}r$ dal centro.

Cosa di richiede

Supponete che la mappa della città sia descritta da un grafo orientato i cui vertici rappresentano gli incroci tra le strade (anche la fine di una strada a fondo chiuso è considerata un incrocio). Gli archi rappresentano tratti di strada, delimitati da due incroci. Ad ogni arco è associato come peso la lunghezza del tratto di strada rappresentato. Il centro della città si trova in un particolare vertice del grafo, indicato con s. Pertanto il valore di r è la massima distanza da s di un vertice del grafo.

Progettate un algoritmo che permetta di individuare gli incroci che si trovano nella fascia centrale o semicentrale dai quali, percorrendo un tratto di strada (senza incroci intermedi), si raggiunga un incrocio delle fascia periferica. Pertanto, l'algoritmo:

- riceve in *input* il grafo orientato pesato $G = (V, E, \omega)$, che descrive la mappa stradale della città, e il vertice $s \in V$ che indica il centro,
- fornisce in *output* un elenco contenente ogni vertice $u \in V$ che rappresenta un incrocio nella fascia centrale o semicentrale, per il quale esiste un arco $(u, v) \in E$, dove v rappresenta un incrocio della fascia periferica.

Risolvete i seguenti punti ***nell'ordine indicato.***

(a) Descrivete sinteticamente a parole la struttura dell'algoritmo ad alto livello. È opportuno ispirarsi a uno degli algoritmi presentati nel corso (indicate quale).

(b) Scrivete l'algoritmo in pseudocodice.

(c) Discutete una possibile rappresentazione del grafo che permetta di implementare facilmente l'algoritmo e fornite una stima del tempo di calcolo in funzione del numero n di vertici e del numero m di archi del grafo.

Le risposte a questo esercizio devono essere adeguatamente giustificate.