Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 3 febbraio 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 la sequenza di numeri `27 34 16 13 18 9 40 6`

(a) Disegnate l'albero AVL che si ottiene inserendo i numeri della sequenza, uno dopo l'altro, nell'ordine indicato, a partire da un albero inizialmente vuoto, ribilanciando quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.

Per (b)(c)(d)(e) fate riferimento all'albero ottenuto in (a)

(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) Considerate l'albero ottenuto al punto (a).
Se NON è *perfettamente bilanciato* disegnate un albero di ricerca perfettamente bilanciato contenente i numeri della sequenza.
Se è *perfettamente bilanciato* scrivete la definizione di albero *perfettamente bilanciato*.

(g) Disegnate l'*albero 2-3* che si ottiene inserendo in un albero 2-3 inizialmente vuoto i numeri della sequenza, uno dopo l'altro, nell'ordine indicato.

(h) Supponete di ordinare la sequenza di numeri fornita inizialmente mediante l'algoritmo `quickSort`, scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di `quickSort`.

2. Siano:

- $\mathcal{A}$ un qualunque problema **NP**-completo,
- $\mathcal{B}$ un qualunque problema **NP**-hard (o **NP**-difficile),
- $\mathcal{C}$ un qualunque problema nella classe **NP**.

Per ognuna delle seguenti affermazioni indicate nel riquadro vicino a essa se è vera o falsa:

- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{A} \notin \mathbf{P}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{C} \notin \mathbf{P}$ ☐
- Se $\mathbf{P} = \mathbf{NP}$ allora $\mathcal{B} \in \mathbf{P}$ ☐
- Se $\mathcal{B} \in \mathbf{P}$ allora $\mathbf{P} = \mathbf{NP}$ ☐

3. In una grande città, dotata di numerose linee di metropolitana che si intersecano in vari punti, vi è un sistema di pagamento dei mezzi pubblici basato sull'uso di carte di credito. La tariffa da pagare in metropolitana *dipende dalla stazione di arrivo e NON da quella di partenza o dal percorso seguito.* Per questa ragione è possibile entrare liberamente nelle stazioni, mentre all'uscita è necessario avvicinare la propria carta di credito ad un apposito dispositivo, che provvede ad addebitare la tariffa relativa alla stazione di uscita.

Il criterio con cui sono stabilite le tariffe da addebitare in uscita dalle singole stazioni è il seguente:

- ad ogni tratto di metropolitana che congiunge direttamente due stazioni (cioè senza stazioni intermedie) è associato un costo,
- vi è una stazione prestabilita chiamata *centro* che si trova al centro della città,
- la tariffa addebitata all'uscita da una stazione x corrisponde al minimo tra i costi dei percorsi dalla stazione *centro* alla stazione x, dove il costo di un percorso è la somma dei costi dei singoli tratti che congiungono le stazioni che lo compongono, *indipendentemente* dalle linee della metropolitana che lo costituiscono (pertanto, le tratte che compongono un percorso potrebbero anche appartenere a diverse linee).

Si osservi che con questo sistema di tariffe, arrivare in una stazione molto lontana dal centro può costare molto, anche se si è percorso un breve tratto. D'altra parte, quando si arriva nella stazione *centro* non si deve pagare nulla, indipendentemente da dove si sia partiti.

Cosa di richiede
Si vuole progettare un algoritmo che, data la mappa della metropolitana con l'indicazione della stazione *centro* e i costi associati ai tratti che collegano direttamente due stazioni, determini le tariffe da pagare in uscita dalle singole stazioni secondo i criteri precedenti.

Le stazioni vengono indicate con gli interi $1, 2, \ldots, n$, dove 1 è la stazione *centro*. Il costo associato al tratto che collega direttamente la stazione i alla stazione j (senza stazioni intermedie), se esiste, è indicato con $\omega(i, j)$.

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

(a) Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.

(b) Ispirandovi a uno degli algoritmi presentati nel corso (indicate quale), progettate un algoritmo che per ogni stazione determini la tariffa addebitata uscendo da essa e scrivetene, ad alto livello, lo pseudocodice.

(c) Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima del tempo di calcolo.

(d) Discutete sinteticamente come estendere l'algoritmo ottenuto al punto (b) in modo che fornisca in output anche un elenco delle stazioni *in ordine di tariffa non decrescente.* Indicate anche come cambia il tempo di calcolo dell'algoritmo (non è richiesto lo pseudocodice, ma solo una descrizione ad alto livello, indicando quale algoritmo di ordinamento utilizzate e in che modo).

Le risposte a questo esercizio devono essere adeguatamente giustificate.

Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 3 febbraio 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 la sequenza di numeri `28 35 17 14 19 10 41 7`

(a) Disegnate l'albero AVL che si ottiene inserendo i numeri della sequenza, uno dopo l'altro, nell'ordine indicato, a partire da un albero inizialmente vuoto, ribilanciando quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.

Per (b)(c)(d)(e) fate riferimento all'albero ottenuto in (a)

(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) Considerate l'albero ottenuto al punto (a).
Se NON è *perfettamente bilanciato* disegnate un albero di ricerca perfettamente bilanciato contenente i numeri della sequenza.
Se è *perfettamente bilanciato* scrivete la definizione di albero *perfettamente bilanciato*.

(g) Disegnate l'*albero 2-3* che si ottiene inserendo in un albero 2-3 inizialmente vuoto i numeri della sequenza, uno dopo l'altro, nell'ordine indicato.

(h) Supponete di ordinare la sequenza di numeri fornita inizialmente mediante l'algoritmo `quickSort`, scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di `quickSort`.

2. Siano:

- $\mathcal{A}$ un qualunque problema **NP**-completo,
- $\mathcal{B}$ un qualunque problema **NP**-hard (o **NP**-difficile),
- $\mathcal{C}$ un qualunque problema nella classe **NP**.

Per ognuna delle seguenti affermazioni indicate nel riquadro vicino a essa se è vera o falsa:

- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{C} \notin \mathbf{P}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{A} \notin \mathbf{P}$ ☐
- Se $\mathcal{B} \in \mathbf{P}$ allora $\mathbf{P} = \mathbf{NP}$ ☐
- Se $\mathbf{P} = \mathbf{NP}$ allora $\mathcal{B} \in \mathbf{P}$ ☐

3. In una grande città, dotata di numerose linee di metropolitana che si intersecano in vari punti, vi è un sistema di pagamento dei mezzi pubblici basato sull'uso di carte di credito. La tariffa da pagare in metropolitana *dipende dalla stazione di arrivo e NON da quella di partenza o dal percorso seguito.* Per questa ragione è possibile entrare liberamente nelle stazioni, mentre all'uscita è necessario avvicinare la propria carta di credito ad un apposito dispositivo, che provvede ad addebitare la tariffa relativa alla stazione di uscita.

Il criterio con cui sono stabilite le tariffe da addebitare in uscita dalle singole stazioni è il seguente:

- ad ogni tratto di metropolitana che congiunge direttamente due stazioni (cioè senza stazioni intermedie) è associato un costo,
- vi è una stazione prestabilita chiamata *centro* che si trova al centro della città,
- la tariffa addebitata all'uscita da una stazione x corrisponde al minimo tra i costi dei percorsi dalla stazione *centro* alla stazione x, dove il costo di un percorso è la somma dei costi dei singoli tratti che congiungono le stazioni che lo compongono, *indipendentemente* dalle linee della metropolitana che lo costituiscono (pertanto, le tratte che compongono un percorso potrebbero anche appartenere a diverse linee).

Si osservi che con questo sistema di tariffe, arrivare in una stazione molto lontana dal centro può costare molto, anche se si è percorso un breve tratto. D'altra parte, quando si arriva nella stazione *centro* non si deve pagare nulla, indipendentemente da dove si sia partiti.

Cosa di richiede
Si vuole progettare un algoritmo che, data la mappa della metropolitana con l'indicazione della stazione *centro* e i costi associati ai tratti che collegano direttamente due stazioni, determini le tariffe da pagare in uscita dalle singole stazioni secondo i criteri precedenti.

Le stazioni vengono indicate con gli interi $1, 2, \ldots, n$, dove 1 è la stazione *centro*. Il costo associato al tratto che collega direttamente la stazione i alla stazione j (senza stazioni intermedie), se esiste, è indicato con $\omega(i, j)$.

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

(a) Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.

(b) Ispirandovi a uno degli algoritmi presentati nel corso (indicate quale), progettate un algoritmo che per ogni stazione determini la tariffa addebitata uscendo da essa e scrivetene, ad alto livello, lo pseudocodice.

(c) Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima del tempo di calcolo.

(d) Discutete sinteticamente come estendere l'algoritmo ottenuto al punto (b) in modo che fornisca in output anche un elenco delle stazioni *in ordine di tariffa non decrescente.* Indicate anche come cambia il tempo di calcolo dell'algoritmo (non è richiesto lo pseudocodice, ma solo una descrizione ad alto livello, indicando quale algoritmo di ordinamento utilizzate e in che modo).

Le risposte a questo esercizio devono essere adeguatamente giustificate.

Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 3 febbraio 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 la sequenza di numeri `29 36 18 15 20 11 42 8`

(a) Disegnate l'albero AVL che si ottiene inserendo i numeri della sequenza, uno dopo l'altro, nell'ordine indicato, a partire da un albero inizialmente vuoto, ribilanciando quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.

Per (b)(c)(d)(e) fate riferimento all'albero ottenuto in (a)

(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) Considerate l'albero ottenuto al punto (a).
Se NON è *perfettamente bilanciato* disegnate un albero di ricerca perfettamente bilanciato contenente i numeri della sequenza.
Se è *perfettamente bilanciato* scrivete la definizione di albero *perfettamente bilanciato*.

(g) Disegnate l'*albero 2-3* che si ottiene inserendo in un albero 2-3 inizialmente vuoto i numeri della sequenza, uno dopo l'altro, nell'ordine indicato.

(h) Supponete di ordinare la sequenza di numeri fornita inizialmente mediante l'algoritmo `quickSort`, scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di `quickSort`.

2. Siano:

- $\mathcal{A}$ un qualunque problema **NP**-completo,
- $\mathcal{B}$ un qualunque problema **NP**-hard (o **NP**-difficile),
- $\mathcal{C}$ un qualunque problema nella classe **NP**.

Per ognuna delle seguenti affermazioni indicate nel riquadro vicino a essa se è vera o falsa:

- Se $\mathbf{P} = \mathbf{NP}$ allora $\mathcal{B} \in \mathbf{P}$ ☐
- Se $\mathcal{B} \in \mathbf{P}$ allora $\mathbf{P} = \mathbf{NP}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{A} \notin \mathbf{P}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{C} \notin \mathbf{P}$ ☐

3. In una grande città, dotata di numerose linee di metropolitana che si intersecano in vari punti, vi è un sistema di pagamento dei mezzi pubblici basato sull'uso di carte di credito. La tariffa da pagare in metropolitana *dipende dalla stazione di arrivo e NON da quella di partenza o dal percorso seguito.* Per questa ragione è possibile entrare liberamente nelle stazioni, mentre all'uscita è necessario avvicinare la propria carta di credito ad un apposito dispositivo, che provvede ad addebitare la tariffa relativa alla stazione di uscita.

Il criterio con cui sono stabilite le tariffe da addebitare in uscita dalle singole stazioni è il seguente:

- ad ogni tratto di metropolitana che congiunge direttamente due stazioni (cioè senza stazioni intermedie) è associato un costo,
- vi è una stazione prestabilita chiamata *centro* che si trova al centro della città,
- la tariffa addebitata all'uscita da una stazione x corrisponde al minimo tra i costi dei percorsi dalla stazione *centro* alla stazione x, dove il costo di un percorso è la somma dei costi dei singoli tratti che congiungono le stazioni che lo compongono, *indipendentemente* dalle linee della metropolitana che lo costituiscono (pertanto, le tratte che compongono un percorso potrebbero anche appartenere a diverse linee).

Si osservi che con questo sistema di tariffe, arrivare in una stazione molto lontana dal centro può costare molto, anche se si è percorso un breve tratto. D'altra parte, quando si arriva nella stazione *centro* non si deve pagare nulla, indipendentemente da dove si sia partiti.

Cosa di richiede
Si vuole progettare un algoritmo che, data la mappa della metropolitana con l'indicazione della stazione *centro* e i costi associati ai tratti che collegano direttamente due stazioni, determini le tariffe da pagare in uscita dalle singole stazioni secondo i criteri precedenti.

Le stazioni vengono indicate con gli interi $1, 2, \ldots, n$, dove 1 è la stazione *centro*. Il costo associato al tratto che collega direttamente la stazione i alla stazione j (senza stazioni intermedie), se esiste, è indicato con $\omega(i, j)$.

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

(a) Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.

(b) Ispirandovi a uno degli algoritmi presentati nel corso (indicate quale), progettate un algoritmo che per ogni stazione determini la tariffa addebitata uscendo da essa e scrivetene, ad alto livello, lo pseudocodice.

(c) Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima del tempo di calcolo.

(d) Discutete sinteticamente come estendere l'algoritmo ottenuto al punto (b) in modo che fornisca in output anche un elenco delle stazioni *in ordine di tariffa non decrescente.* Indicate anche come cambia il tempo di calcolo dell'algoritmo (non è richiesto lo pseudocodice, ma solo una descrizione ad alto livello, indicando quale algoritmo di ordinamento utilizzate e in che modo).

Le risposte a questo esercizio devono essere adeguatamente giustificate.

Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 3 febbraio 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 la sequenza di numeri `30 37 19 16 21 12 43 9`

(a) Disegnate l'albero AVL che si ottiene inserendo i numeri della sequenza, uno dopo l'altro, nell'ordine indicato, a partire da un albero inizialmente vuoto, ribilanciando quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.

Per (b)(c)(d)(e) fate riferimento all'albero ottenuto in (a)

(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) Considerate l'albero ottenuto al punto (a).
Se NON è *perfettamente bilanciato* disegnate un albero di ricerca perfettamente bilanciato contenente i numeri della sequenza.
Se è *perfettamente bilanciato* scrivete la definizione di albero *perfettamente bilanciato*.

(g) Disegnate l'*albero 2-3* che si ottiene inserendo in un albero 2-3 inizialmente vuoto i numeri della sequenza, uno dopo l'altro, nell'ordine indicato.

(h) Supponete di ordinare la sequenza di numeri fornita inizialmente mediante l'algoritmo `quickSort`, scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di `quickSort`.

2. Siano:

- $\mathcal{A}$ un qualunque problema **NP**-completo,
- $\mathcal{B}$ un qualunque problema **NP**-hard (o **NP**-difficile),
- $\mathcal{C}$ un qualunque problema nella classe **NP**.

Per ognuna delle seguenti affermazioni indicate nel riquadro vicino a essa se è vera o falsa:

- Se $\mathcal{B} \in \mathbf{P}$ allora $\mathbf{P} = \mathbf{NP}$ ☐
- Se $\mathbf{P} = \mathbf{NP}$ allora $\mathcal{B} \in \mathbf{P}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{C} \notin \mathbf{P}$ ☐
- Se $\mathbf{P} \neq \mathbf{NP}$ allora $\mathcal{A} \notin \mathbf{P}$ ☐

3. In una grande città, dotata di numerose linee di metropolitana che si intersecano in vari punti, vi è un sistema di pagamento dei mezzi pubblici basato sull'uso di carte di credito. La tariffa da pagare in metropolitana *dipende dalla stazione di arrivo e NON da quella di partenza o dal percorso seguito.* Per questa ragione è possibile entrare liberamente nelle stazioni, mentre all'uscita è necessario avvicinare la propria carta di credito ad un apposito dispositivo, che provvede ad addebitare la tariffa relativa alla stazione di uscita.

Il criterio con cui sono stabilite le tariffe da addebitare in uscita dalle singole stazioni è il seguente:

- ad ogni tratto di metropolitana che congiunge direttamente due stazioni (cioè senza stazioni intermedie) è associato un costo,
- vi è una stazione prestabilita chiamata *centro* che si trova al centro della città,
- la tariffa addebitata all'uscita da una stazione x corrisponde al minimo tra i costi dei percorsi dalla stazione *centro* alla stazione x, dove il costo di un percorso è la somma dei costi dei singoli tratti che congiungono le stazioni che lo compongono, *indipendentemente* dalle linee della metropolitana che lo costituiscono (pertanto, le tratte che compongono un percorso potrebbero anche appartenere a diverse linee).

Si osservi che con questo sistema di tariffe, arrivare in una stazione molto lontana dal centro può costare molto, anche se si è percorso un breve tratto. D'altra parte, quando si arriva nella stazione *centro* non si deve pagare nulla, indipendentemente da dove si sia partiti.

Cosa di richiede
Si vuole progettare un algoritmo che, data la mappa della metropolitana con l'indicazione della stazione *centro* e i costi associati ai tratti che collegano direttamente due stazioni, determini le tariffe da pagare in uscita dalle singole stazioni secondo i criteri precedenti.

Le stazioni vengono indicate con gli interi $1, 2, \ldots, n$, dove 1 è la stazione *centro*. Il costo associato al tratto che collega direttamente la stazione i alla stazione j (senza stazioni intermedie), se esiste, è indicato con $\omega(i, j)$.

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

(a) Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.

(b) Ispirandovi a uno degli algoritmi presentati nel corso (indicate quale), progettate un algoritmo che per ogni stazione determini la tariffa addebitata uscendo da essa e scrivetene, ad alto livello, lo pseudocodice.

(c) Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima del tempo di calcolo.

(d) Discutete sinteticamente come estendere l'algoritmo ottenuto al punto (b) in modo che fornisca in output anche un elenco delle stazioni *in ordine di tariffa non decrescente.* Indicate anche come cambia il tempo di calcolo dell'algoritmo (non è richiesto lo pseudocodice, ma solo una descrizione ad alto livello, indicando quale algoritmo di ordinamento utilizzate e in che modo).

Le risposte a questo esercizio devono essere adeguatamente giustificate.