

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 24 febbraio 2017

Matricola.....

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1, 2, 3 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 4 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su tutto ciò che consegnate.

1. Considerate un albero binario di ricerca e un albero AVL ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 15 12 10 7 11 9 13

(a) Disegnate l'albero binario di ricerca	(b) Disegnate l'albero AVL
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine anticipato dell'albero AVL	(f) L'albero AVL ottenuto al punto (b) è perfettamente bilanciato? <input type="checkbox"/> (g) Scrivete la definizione di albero AVL.
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine simmetrico dell'albero AVL	
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine posticipato dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 8 4 15 12 10 7 11 9 13

(a) Supponete di ordinare la sequenza 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 .
(b) Supponete di ordinare la sequenza mediante l'algoritmo heapSort . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(c) Supponete di ordinare la sequenza mediante l'algoritmo bubbleSort . Indicate il contenuto dell'array dopo le prime due iterazioni del ciclo principale dell'algoritmo.

3. Nel riquadro che segue ciascuna affermazione, scrivete V se l'affermazione è vera, F se è falsa:

- | | |
|---|--------------------------|
| (a) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , effettuando $O(n \log n)$ confronti. | <input type="checkbox"/> |
| (b) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , in tempo $O(n \log n)$. | <input type="checkbox"/> |
| (c) Per determinare l'elemento minimo in un array ordinato contenente n elementi è necessario, in media, tempo logaritmico in n . | <input type="checkbox"/> |
| (d) L'elemento minimo in un albero AVL contenente n nodi può essere determinato in tempo logaritmico in n . | <input type="checkbox"/> |
| (e) Il problema di determinare il cammino minimo tra una coppia di vertici in un grafo orientato, con pesi sugli archi, è NP-completo (ipotizzando $P \neq NP$). | <input type="checkbox"/> |
| (f) L'algoritmo quickSort ordina un array di n elementi senza utilizzare spazio di memoria aggiuntivo. | <input type="checkbox"/> |
| (g) L'algoritmo quickSort è, in ogni caso, il più veloce algoritmo di ordinamento. | <input type="checkbox"/> |
| (h) Dato un problema di ottimizzazione, utilizzando l'algoritmo greedy è sempre possibile determinarne in tempo polinomiale la soluzione ottima. | <input type="checkbox"/> |
| (i) Il merge di due vettori di lunghezza n può essere effettuato utilizzando $n - 1$ confronti. | <input type="checkbox"/> |
| (j) Per risolvere velocemente il problema di soddisfacibilità è indispensabile utilizzare computer estremamente veloci. | <input type="checkbox"/> |

4. In una piccola città si è deciso di creare un servizio di posta pneumatica con cui, da una centrale di invio, sia possibile distribuire pressoché istantaneamente piccoli pacchi verso punti di ricezione. Il luogo in cui collocare la centrale è stato selezionato all'incrocio tra alcune strade della città. I punti di ricezione saranno collocati in tutti gli altri incroci. I tubi utilizzati dal sistema verranno posti sotto alcuni tratti di strada della città in modo che ciascun punto di ricezione sia collegato alla centrale di invio mediante un tubo dedicato. Al fine di ridurre i costi di realizzazione del progetto, è necessario trovare una soluzione che minimizzi la lunghezza totale dei tubi utilizzati.

- Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.
- Progettate un algoritmo che determini, per ogni punto di ricezione, la lunghezza del tubo che lo collega alla centrale di invio: descrivete l'algoritmo a parole e poi, ad alto livello, in pseudocodice. *È opportuno ispirarsi a uno degli algoritmi presentati nel corso (indicate quale).*
- Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima dei tempi di calcolo.

Note:

- Un tratto di strada è una parte di strada delimitata da due incroci e senza incroci all'interno. La fine di una strada a fondo chiuso è considerata un incrocio.
- Ciascun tubo è dedicato al collegamento tra la centrale di invio e uno specifico punto di distribuzione. Pertanto sotto uno stesso tratto di strada possono essere collocati più tubi.

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 24 febbraio 2017

Matricola.....

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1, 2, 3 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 4 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su tutto ciò che consegnate.

1. Considerate un albero binario di ricerca e un albero AVL ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 13 10 8 5 9 7 11

(a) Disegnate l'albero binario di ricerca	(b) Disegnate l'albero AVL
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine anticipato dell'albero AVL	(f) L'albero AVL ottenuto al punto (b) è perfettamente bilanciato? <input type="checkbox"/> (g) Scrivete la definizione di albero AVL.
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine simmetrico dell'albero AVL	
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine posticipato dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 6 2 13 10 8 5 9 7 11

(a) Supponete di ordinare la sequenza mediante l'algoritmo <code>quickSort</code> , scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di <code>quickSort</code> .
(b) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>bubbleSort</code> . Indicate il contenuto dell'array dopo le prime due iterazioni del ciclo principale dell'algoritmo.

3. Nel riquadro che segue ciascuna affermazione, scrivete V se l'affermazione è vera, F se è falsa:

- (a) Per determinare l'elemento minimo in un array ordinato contenente n elementi è necessario, in media, tempo logaritmico in n .
- (b) L'elemento minimo in un albero AVL contenente n nodi può essere determinato in tempo logaritmico in n .
- (c) Il problema di determinare il cammino minimo tra una coppia di vertici in un grafo orientato, con pesi sugli archi, è NP-completo (ipotizzando $P \neq NP$).
- (d) Per risolvere velocemente il problema di soddisfacibilità è indispensabile utilizzare computer estremamente veloci.
- (e) Utilizzando l'algoritmo `heapSort` è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , effettuando $O(n \log n)$ confronti.
- (f) Utilizzando l'algoritmo `heapSort` è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , in tempo $O(n \log n)$.
- (g) L'algoritmo `quickSort` ordina un array di n elementi senza utilizzare spazio di memoria aggiuntivo.
- (h) L'algoritmo `quickSort` è, in ogni caso, il più veloce algoritmo di ordinamento.
- (i) Dato un problema di ottimizzazione, utilizzando l'algoritmo greedy è sempre possibile determinarne in tempo polinomiale la soluzione ottima.
- (j) Il merge di due vettori di lunghezza n può essere effettuato utilizzando $n - 1$ confronti.

4. In una piccola città si è deciso di creare un servizio di posta pneumatica con cui, da una centrale di invio, sia possibile distribuire pressoché istantaneamente piccoli pacchi verso punti di ricezione. Il luogo in cui collocare la centrale è stato selezionato all'incrocio tra alcune strade della città. I punti di ricezione saranno collocati in tutti gli altri incroci. I tubi utilizzati dal sistema verranno posti sotto alcuni tratti di strada della città in modo che ciascun punto di ricezione sia collegato alla centrale di invio mediante un tubo dedicato. Al fine di ridurre i costi di realizzazione del progetto, è necessario trovare una soluzione che minimizzi la lunghezza totale dei tubi utilizzati.

- (a) Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.
- (b) Progettate un algoritmo che determini, per ogni punto di ricezione, la lunghezza del tubo che lo collega alla centrale di invio: descrivete l'algoritmo a parole e poi, ad alto livello, in pseudocodice. *È opportuno ispirarsi a uno degli algoritmi presentati nel corso (indicate quale).*
- (c) Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima dei tempi di calcolo.

Note:

- Un tratto di strada è una parte di strada delimitata da due incroci e senza incroci all'interno. La fine di una strada a fondo chiuso è considerata un incrocio.
- Ciascun tubo è dedicato al collegamento tra la centrale di invio e uno specifico punto di distribuzione. Pertanto sotto uno stesso tratto di strada possono essere collocati più tubi.

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 24 febbraio 2017

Matricola.....

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1, 2, 3 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 4 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su tutto ciò che consegnate.

1. Considerate un albero binario di ricerca e un albero AVL ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 17 14 12 9 13 11 15

(a) Disegnate l'albero binario di ricerca	(b) Disegnate l'albero AVL
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine anticipato dell'albero AVL	(f) L'albero AVL ottenuto al punto (b) è perfettamente bilanciato? <input type="checkbox"/> (g) Scrivete la definizione di albero AVL.
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine simmetrico dell'albero AVL	
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine posticipato dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 10 6 17 14 12 9 13 11 15

(a) Supponete di ordinare la sequenza mediante l'algoritmo <code>quickSort</code> , scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di <code>quickSort</code> .
(b) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>bubbleSort</code> . Indicate il contenuto dell'array dopo le prime due iterazioni del ciclo principale dell'algoritmo.

3. Nel riquadro che segue ciascuna affermazione, scrivete V se l'affermazione è vera, F se è falsa:

- | | |
|--|--------------------------|
| (a) Il problema di determinare il cammino minimo tra una coppia di vertici in un grafo orientato, con pesi sugli archi, è NP-completo (ipotizzando $P \neq NP$). | <input type="checkbox"/> |
| (b) Dato un problema di ottimizzazione, utilizzando l'algoritmo greedy è sempre possibile determinarne in tempo polinomiale la soluzione ottima. | <input type="checkbox"/> |
| (c) Per determinare l'elemento minimo in un array ordinato contenente n elementi è necessario, in media, tempo logaritmico in n . | <input type="checkbox"/> |
| (d) L'elemento minimo in un albero AVL contenente n nodi può essere determinato in tempo logaritmico in n . | <input type="checkbox"/> |
| (e) L'algoritmo quickSort ordina un array di n elementi senza utilizzare spazio di memoria aggiuntivo. | <input type="checkbox"/> |
| (f) L'algoritmo quickSort è, in ogni caso, il più veloce algoritmo di ordinamento. | <input type="checkbox"/> |
| (g) Il merge di due vettori di lunghezza n può essere effettuato utilizzando $n - 1$ confronti. | <input type="checkbox"/> |
| (h) Per risolvere velocemente il problema di soddisfacibilità è indispensabile utilizzare computer estremamente veloci. | <input type="checkbox"/> |
| (i) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , effettuando $O(n \log n)$ confronti. | <input type="checkbox"/> |
| (j) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , in tempo $O(n \log n)$. | <input type="checkbox"/> |

4. In una piccola città si è deciso di creare un servizio di posta pneumatica con cui, da una centrale di invio, sia possibile distribuire pressoché istantaneamente piccoli pacchi verso punti di ricezione. Il luogo in cui collocare la centrale è stato selezionato all'incrocio tra alcune strade della città. I punti di ricezione saranno collocati in tutti gli altri incroci. I tubi utilizzati dal sistema verranno posti sotto alcuni tratti di strada della città in modo che ciascun punto di ricezione sia collegato alla centrale di invio mediante un tubo dedicato. Al fine di ridurre i costi di realizzazione del progetto, è necessario trovare una soluzione che minimizzi la lunghezza totale dei tubi utilizzati.

- Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.
- Progettate un algoritmo che determini, per ogni punto di ricezione, la lunghezza del tubo che lo collega alla centrale di invio: descrivete l'algoritmo a parole e poi, ad alto livello, in pseudocodice. *È opportuno ispirarsi a uno degli algoritmi presentati nel corso (indicate quale).*
- Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima dei tempi di calcolo.

Note:

- Un tratto di strada è una parte di strada delimitata da due incroci e senza incroci all'interno. La fine di una strada a fondo chiuso è considerata un incrocio.
- Ciascun tubo è dedicato al collegamento tra la centrale di invio e uno specifico punto di distribuzione. Pertanto sotto uno stesso tratto di strada possono essere collocati più tubi.

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 24 febbraio 2017

Matricola.....

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1, 2, 3 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 4 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su tutto ciò che consegnate.

1. Considerate un albero binario di ricerca e un albero AVL ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 11 8 6 3 7 5 9

(a) Disegnate l'albero binario di ricerca	(b) Disegnate l'albero AVL
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine anticipato dell'albero AVL	(f) L'albero AVL ottenuto al punto (b) è perfettamente bilanciato? <input type="checkbox"/> (g) Scrivete la definizione di albero AVL.
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine simmetrico dell'albero AVL	
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in ordine posticipato dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 4 1 11 8 6 3 7 5 9

(a) Supponete di ordinare la sequenza 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 .
(b) Supponete di ordinare la sequenza mediante l'algoritmo heapSort . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(c) Supponete di ordinare la sequenza mediante l'algoritmo bubbleSort . Indicate il contenuto dell'array dopo le prime due iterazioni del ciclo principale dell'algoritmo.

3. Nel riquadro che segue ciascuna affermazione, scrivete V se l'affermazione è vera, F se è falsa:

- | | |
|--|--------------------------|
| (a) Per risolvere velocemente il problema di soddisfacibilità è indispensabile utilizzare computer estremamente veloci. | <input type="checkbox"/> |
| (b) L'algoritmo quickSort ordina un array di n elementi senza utilizzare spazio di memoria aggiuntivo. | <input type="checkbox"/> |
| (c) L'algoritmo quickSort è, in ogni caso, il più veloce algoritmo di ordinamento. | <input type="checkbox"/> |
| (d) Il merge di due vettori di lunghezza n può essere effettuato utilizzando $n - 1$ confronti. | <input type="checkbox"/> |
| (e) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , effettuando $O(n \log n)$ confronti. | <input type="checkbox"/> |
| (f) Dato un problema di ottimizzazione, utilizzando l'algoritmo greedy è sempre possibile determinarne in tempo polinomiale la soluzione ottima. | <input type="checkbox"/> |
| (g) Utilizzando l'algoritmo heapSort è possibile ordinare un array contenente n stringhe, ognuna delle quali di lunghezza n , in tempo $O(n \log n)$. | <input type="checkbox"/> |
| (h) Per determinare l'elemento minimo in un array ordinato contenente n elementi è necessario, in media, tempo logaritmico in n . | <input type="checkbox"/> |
| (i) L'elemento minimo in un albero AVL contenente n nodi può essere determinato in tempo logaritmico in n . | <input type="checkbox"/> |
| (j) Il problema di determinare il cammino minimo tra una coppia di vertici in un grafo orientato, con pesi sugli archi, è NP-completo (ipotizzando $P \neq NP$). | <input type="checkbox"/> |

4. In una piccola città si è deciso di creare un servizio di posta pneumatica con cui, da una centrale di invio, sia possibile distribuire pressoché istantaneamente piccoli pacchi verso punti di ricezione. Il luogo in cui collocare la centrale è stato selezionato all'incrocio tra alcune strade della città. I punti di ricezione saranno collocati in tutti gli altri incroci. I tubi utilizzati dal sistema verranno posti sotto alcuni tratti di strada della città in modo che ciascun punto di ricezione sia collegato alla centrale di invio mediante un tubo dedicato. Al fine di ridurre i costi di realizzazione del progetto, è necessario trovare una soluzione che minimizzi la lunghezza totale dei tubi utilizzati.

- Spiegate come il problema possa essere descritto e formalizzato in termini di grafi.
- Progettate un algoritmo che determini, per ogni punto di ricezione, la lunghezza del tubo che lo collega alla centrale di invio: descrivete l'algoritmo a parole e poi, ad alto livello, in pseudocodice. *È opportuno ispirarsi a uno degli algoritmi presentati nel corso (indicate quale).*
- Discutete una possibile rappresentazione del grafo utilizzato che permetta di implementare facilmente l'algoritmo e fornite una stima dei tempi di calcolo.

Note:

- Un tratto di strada è una parte di strada delimitata da due incroci e senza incroci all'interno. La fine di una strada a fondo chiuso è considerata un incrocio.
- Ciascun tubo è dedicato al collegamento tra la centrale di invio e uno specifico punto di distribuzione. Pertanto sotto uno stesso tratto di strada possono essere collocati più tubi.