

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 18 gennaio 2019

Matricola.....

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 binario di ricerca ottenuto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da un albero inizialmente vuoto, senza effettuare operazioni di bilanciamento:

20 25 15 17 19 14 22 12 30

(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 AVL?

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

(g) Se al punto (f) avete risposto SI scrivete la definizione di albero AVL. Se avete risposto NO disegnate l'albero AVL che si ottiene inserendo la stessa sequenza di numeri in un albero inizialmente vuoto, ribilanciando, quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.	(i) Se al punto (h) avete risposto SI scrivete la definizione di albero perfettamente bilanciato. Se avete risposto NO disegnate un qualunque albero di ricerca perfettamente bilanciato contenente gli stessi numeri
---	--

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:
20 25 24 17 21 14 22 12

(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>mergeSort</code> . Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive, prima del <code>merge</code> finale.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(d) Supponete di ordinare la sequenza mediante l'algoritmo <code>insertionSort</code> . Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.

3. Dato un vettore V di interi, un *sottovettore monotono* è una qualunque porzione di V i cui elementi siano ordinati in modo non decrescente o non crescente. Si vuole costruire un algoritmo che, dato un vettore V di *interi non negativi*, determini:

- il valore massimo max delle somme dei suoi sottovettori monotoni,
- l'elenco di tutti i sottovettori monotoni che hanno somma uguale a max .

Esempio

Dato il vettore $V = (3, 5, 7, 6, 5, 1, 2, 3, 3, 4)$, i sottovettori $(3, 5, 7)$, $(7, 6, 5, 1)$, $(1, 2, 3, 3, 4)$ sono monotoni e massimali (cioè se prolungati con gli elementi adiacenti producono vettori non monotoni). Le loro somme sono rispettivamente 15, 19, 13. Pertanto l'algoritmo dovrà produrre in output 19 e il sottovettore $(7, 6, 5, 1)$.

Nel caso il vettore sia $(3, 5, 7, 4, 3, 1, 2, 3, 3, 4)$, l'algoritmo dovrà produrre in output 15 e i sottovettori $(3, 5, 7)$, $(7, 4, 3, 1)$.

Cosa di richiede

Dato un vettore $V = (v_1, v_2, \dots, v_n)$ contenente n numeri interi positivi, per $i = 1, \dots, n$ indichiamo con c_i la somma del più lungo sottovettore monotono di V che termina in posizione i . Sia C il vettore (c_1, c_2, \dots, c_n) (nel primo esempio si ha $C = (3, 8, 15, 13, 18, 19, 3, 6, 9, 13)$).

Risolvete i seguenti punti *nell'ordine indicato*.

- Scrivete una o più formule che permettano di ricavare il generico valore c_i . Per $i > 1$, c_i deve essere calcolato in funzione di c_{i-1} , di v_i e, se utile, di altri valori di indice $< i$ nel vettore V o di altre informazioni relative alla parte di vettore V fino alla posizione i .
- Indicate come dal vettore C sia possibile ottenere il risultato al punto (i).
- Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo il vettore V , calcoli il vettore C e fornisca il risultato al punto (i) (utilizzate quanto scritto ai punti (a) e (b)).
- Fornite una stima in funzione di n del tempo totale utilizzato dalla parte di algoritmo scritta al punto (c).
- Descrivete *sinteticamente a parole* come ricavare dal vettore C quanto richiesto al punto (ii).
- Fornite una stima del tempo totale utilizzato dalla parte di algoritmo descritta al punto (e) in funzione di n .

Note

- Le risposte devono essere adeguatamente giustificate.
- Non cambiate i nomi stabiliti nel testo dell'esercizio, in particolare i nomi dei vettori V e C .

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 18 gennaio 2019

Matricola.....

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 binario di ricerca ottenuto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da un albero inizialmente vuoto, senza effettuare operazioni di bilanciamento:

21 26 16 18 20 15 23 13 31

(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 AVL?

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

(g) Se al punto (f) avete risposto SI scrivete la definizione di albero AVL. Se avete risposto NO disegnate l'albero AVL che si ottiene inserendo la stessa sequenza di numeri in un albero inizialmente vuoto, ribilanciando, quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.	(i) Se al punto (h) avete risposto SI scrivete la definizione di albero perfettamente bilanciato. Se avete risposto NO disegnate un qualunque albero di ricerca perfettamente bilanciato contenente gli stessi numeri
---	--

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:

21 26 25 18 22 15 23 13

(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>mergeSort</code> . Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive, prima del <code>merge</code> finale.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(d) Supponete di ordinare la sequenza mediante l'algoritmo <code>insertionSort</code> . Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.

3. Dato un vettore V di interi, un *sottovettore monotono* è una qualunque porzione di V i cui elementi siano ordinati in modo non decrescente o non crescente. Si vuole costruire un algoritmo che, dato un vettore V di *interi non negativi*, determini:

- il valore massimo max delle somme dei suoi sottovettori monotoni,
- l'elenco di tutti i sottovettori monotoni che hanno somma uguale a max .

Esempio

Dato il vettore $V = (3, 5, 7, 6, 5, 1, 2, 3, 3, 4)$, i sottovettori $(3, 5, 7)$, $(7, 6, 5, 1)$, $(1, 2, 3, 3, 4)$ sono monotoni e massimali (cioè se prolungati con gli elementi adiacenti producono vettori non monotoni). Le loro somme sono rispettivamente 15, 19, 13. Pertanto l'algoritmo dovrà produrre in output 19 e il sottovettore $(7, 6, 5, 1)$.

Nel caso il vettore sia $(3, 5, 7, 4, 3, 1, 2, 3, 3, 4)$, l'algoritmo dovrà produrre in output 15 e i sottovettori $(3, 5, 7)$, $(7, 4, 3, 1)$.

Cosa di richiede

Dato un vettore $V = (v_1, v_2, \dots, v_n)$ contenente n numeri interi positivi, per $i = 1, \dots, n$ indichiamo con c_i la somma del più lungo sottovettore monotono di V che termina in posizione i . Sia C il vettore (c_1, c_2, \dots, c_n) (nel primo esempio si ha $C = (3, 8, 15, 13, 18, 19, 3, 6, 9, 13)$).

Risolvete i seguenti punti *nell'ordine indicato*.

- Scrivete una o più formule che permettano di ricavare il generico valore c_i . Per $i > 1$, c_i deve essere calcolato in funzione di c_{i-1} , di v_i e, se utile, di altri valori di indice $< i$ nel vettore V o di altre informazioni relative alla parte di vettore V fino alla posizione i .
- Indicate come dal vettore C sia possibile ottenere il risultato al punto (i).
- Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo il vettore V , calcoli il vettore C e fornisca il risultato al punto (i) (utilizzate quanto scritto ai punti (a) e (b)).
- Fornite una stima in funzione di n del tempo totale utilizzato dalla parte di algoritmo scritta al punto (c).
- Descrivete *sinteticamente a parole* come ricavare dal vettore C quanto richiesto al punto (ii).
- Fornite una stima del tempo totale utilizzato dalla parte di algoritmo descritta al punto (e) in funzione di n .

Note

- Le risposte devono essere adeguatamente giustificate.
- Non cambiate i nomi stabiliti nel testo dell'esercizio, in particolare i nomi dei vettori V e C .

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 18 gennaio 2019

Matricola.....

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 binario di ricerca ottenuto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da un albero inizialmente vuoto, senza effettuare operazioni di bilanciamento:

22 27 17 19 21 16 24 14 32

(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 AVL?

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

(g) Se al punto (f) avete risposto SI scrivete la definizione di albero AVL. Se avete risposto NO disegnate l'albero AVL che si ottiene inserendo la stessa sequenza di numeri in un albero inizialmente vuoto, ribilanciando, quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.	(i) Se al punto (h) avete risposto SI scrivete la definizione di albero perfettamente bilanciato. Se avete risposto NO disegnate un qualunque albero di ricerca perfettamente bilanciato contenente gli stessi numeri
---	--

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:

22 27 26 19 23 16 24 14

(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>mergeSort</code> . Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive, prima del <code>merge</code> finale.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(d) Supponete di ordinare la sequenza mediante l'algoritmo <code>insertionSort</code> . Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.

3. Dato un vettore V di interi, un *sottovettore monotono* è una qualunque porzione di V i cui elementi siano ordinati in modo non decrescente o non crescente. Si vuole costruire un algoritmo che, dato un vettore V di *interi non negativi*, determini:

- il valore massimo max delle somme dei suoi sottovettori monotoni,
- l'elenco di tutti i sottovettori monotoni che hanno somma uguale a max .

Esempio

Dato il vettore $V = (3, 5, 7, 6, 5, 1, 2, 3, 3, 4)$, i sottovettori $(3, 5, 7)$, $(7, 6, 5, 1)$, $(1, 2, 3, 3, 4)$ sono monotoni e massimali (cioè se prolungati con gli elementi adiacenti producono vettori non monotoni). Le loro somme sono rispettivamente 15, 19, 13. Pertanto l'algoritmo dovrà produrre in output 19 e il sottovettore $(7, 6, 5, 1)$.

Nel caso il vettore sia $(3, 5, 7, 4, 3, 1, 2, 3, 3, 4)$, l'algoritmo dovrà produrre in output 15 e i sottovettori $(3, 5, 7)$, $(7, 4, 3, 1)$.

Cosa di richiede

Dato un vettore $V = (v_1, v_2, \dots, v_n)$ contenente n numeri interi positivi, per $i = 1, \dots, n$ indichiamo con c_i la somma del più lungo sottovettore monotono di V che termina in posizione i . Sia C il vettore (c_1, c_2, \dots, c_n) (nel primo esempio si ha $C = (3, 8, 15, 13, 18, 19, 3, 6, 9, 13)$).

Risolvete i seguenti punti *nell'ordine indicato*.

- Scrivete una o più formule che permettano di ricavare il generico valore c_i . Per $i > 1$, c_i deve essere calcolato in funzione di c_{i-1} , di v_i e, se utile, di altri valori di indice $< i$ nel vettore V o di altre informazioni relative alla parte di vettore V fino alla posizione i .
- Indicate come dal vettore C sia possibile ottenere il risultato al punto (i).
- Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo il vettore V , calcoli il vettore C e fornisca il risultato al punto (i) (utilizzate quanto scritto ai punti (a) e (b)).
- Fornite una stima in funzione di n del tempo totale utilizzato dalla parte di algoritmo scritta al punto (c).
- Descrivete *sinteticamente a parole* come ricavare dal vettore C quanto richiesto al punto (ii).
- Fornite una stima del tempo totale utilizzato dalla parte di algoritmo descritta al punto (e) in funzione di n .

Note

- Le risposte devono essere adeguatamente giustificate.
- Non cambiate i nomi stabiliti nel testo dell'esercizio, in particolare i nomi dei vettori V e C .

Cognome.....

Algoritmi e Strutture Dati

Nome.....

Prova scritta del 18 gennaio 2019

Matricola.....

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 binario di ricerca ottenuto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da un albero inizialmente vuoto, senza effettuare operazioni di bilanciamento:

23 28 18 20 22 17 25 15 33

(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 AVL?

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

(g) Se al punto (f) avete risposto SI scrivete la definizione di albero AVL. Se avete risposto NO disegnate l'albero AVL che si ottiene inserendo la stessa sequenza di numeri in un albero inizialmente vuoto, ribilanciando, quando necessario, in modo da ottenere un albero AVL dopo ogni inserimento.	(i) Se al punto (h) avete risposto SI scrivete la definizione di albero perfettamente bilanciato. Se avete risposto NO disegnate un qualunque albero di ricerca perfettamente bilanciato contenente gli stessi numeri
---	--

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:

23 28 27 20 24 17 25 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>mergeSort</code> . Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive, prima del <code>merge</code> finale.
(c) Supponete di ordinare la sequenza mediante l'algoritmo <code>heapSort</code> . Indicate il contenuto dell'array dopo averlo trasformato in uno heap.
(d) Supponete di ordinare la sequenza mediante l'algoritmo <code>insertionSort</code> . Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.

3. Dato un vettore V di interi, un *sottovettore monotono* è una qualunque porzione di V i cui elementi siano ordinati in modo non decrescente o non crescente. Si vuole costruire un algoritmo che, dato un vettore V di *interi non negativi*, determini:

- il valore massimo max delle somme dei suoi sottovettori monotoni,
- l'elenco di tutti i sottovettori monotoni che hanno somma uguale a max .

Esempio

Dato il vettore $V = (3, 5, 7, 6, 5, 1, 2, 3, 3, 4)$, i sottovettori $(3, 5, 7)$, $(7, 6, 5, 1)$, $(1, 2, 3, 3, 4)$ sono monotoni e massimali (cioè se prolungati con gli elementi adiacenti producono vettori non monotoni). Le loro somme sono rispettivamente 15, 19, 13. Pertanto l'algoritmo dovrà produrre in output 19 e il sottovettore $(7, 6, 5, 1)$.

Nel caso il vettore sia $(3, 5, 7, 4, 3, 1, 2, 3, 3, 4)$, l'algoritmo dovrà produrre in output 15 e i sottovettori $(3, 5, 7)$, $(7, 4, 3, 1)$.

Cosa di richiede

Dato un vettore $V = (v_1, v_2, \dots, v_n)$ contenente n numeri interi positivi, per $i = 1, \dots, n$ indichiamo con c_i la somma del più lungo sottovettore monotono di V che termina in posizione i . Sia C il vettore (c_1, c_2, \dots, c_n) (nel primo esempio si ha $C = (3, 8, 15, 13, 18, 19, 3, 6, 9, 13)$).

Risolvete i seguenti punti *nell'ordine indicato*.

- Scrivete una o più formule che permettano di ricavare il generico valore c_i . Per $i > 1$, c_i deve essere calcolato in funzione di c_{i-1} , di v_i e, se utile, di altri valori di indice $< i$ nel vettore V o di altre informazioni relative alla parte di vettore V fino alla posizione i .
- Indicate come dal vettore C sia possibile ottenere il risultato al punto (i).
- Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo il vettore V , calcoli il vettore C e fornisca il risultato al punto (i) (utilizzate quanto scritto ai punti (a) e (b)).
- Fornite una stima in funzione di n del tempo totale utilizzato dalla parte di algoritmo scritta al punto (c).
- Descrivete *sinteticamente a parole* come ricavare dal vettore C quanto richiesto al punto (ii).
- Fornite una stima del tempo totale utilizzato dalla parte di algoritmo descritta al punto (e) in funzione di n .

Note

- Le risposte devono essere adeguatamente giustificate.
- Non cambiate i nomi stabiliti nel testo dell'esercizio, in particolare i nomi dei vettori V e C .