

Cognome.....

# Algoritmi e Strutture Dati

Nome.....

Prova scritta del 16 giugno 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 AVL e un albero 2-3 ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 16 22 28 30 14 15 12

(a) Disegnate l'albero AVL	(b) Disegnate l'albero 2-3
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ampiezza</i> dell'albero AVL	(f) L'albero AVL ottenuto al punto (a) è perfettamente bilanciato? <input type="checkbox"/>
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ordine anticipato</i> dell'albero AVL	(g) Se avete risposto NO al punto (f), disegnate un albero <i>perfettamente bilanciato</i> contenente gli stessi numeri
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ordine posticipato</i> dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 555 323 434 626 543 727 343 929

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

3. Considerate funzione  $f : \{a, b, \dots, z\} \rightarrow \{0, 1, \dots, 15\}$  definita come segue:

$x$	$f(x)$
a	0
b	1
c	2
d	3
e	4
f	5
g	6
h	6
i	7

$x$	$f(x)$
j	7
k	7
l	8
m	9
n	10
o	10
p	11
q	12
r	12

$x$	$f(x)$
s	13
t	13
u	14
v	14
w	15
x	15
y	15
z	15

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Sia  $h$  la funzione che trasforma ogni parola  $k$  sull'alfabeto  $\{a, b, \dots, z\}$  nell'intero che si ottiene applicando  $f$  al primo carattere di  $k$  e  $g$  la funzione che trasforma ogni parola nel più piccolo numero primo maggiore o uguale al numero di vocali presenti in  $k$ .

Esempi:  $h(\text{gatto}) = 6$ ,  $g(\text{gatto}) = 2$ ,  $h(\text{lucertola}) = 8$ ,  $g(\text{lucertola}) = 5$ .

Inserite nella tabella hash a destra, inizialmente vuota, le seguenti parole, nell'ordine indicato:

mulo dingo giraffa lepre tigre dromedario elefante  
cane

Come funzione hash utilizzate  $h$ . Per la gestione delle collisioni utilizzate l'hashing doppio mediante la funzione

$$c(k, i) = (h(k) + i \cdot g(k)) \bmod 16$$

4. L'eccentrico sindaco di una città di cui non si può riportare il nome, ha fatto costruire un sistema di aromaterapia che, utilizzando una rete di tubazioni, dovrebbe diffondere in tutte le abitazioni sostanze aromatiche gradevoli e stimolanti. Il sistema è stato messo in funzione da qualche settimana.

Dopo un breve periodo di grande euforia, gli abitanti della città hanno cominciato a mostrare forti segni di depressione e stanchezza. Gli psicologi che sono stati interpellati per studiare la situazione sono giunti alla conclusione che ciò sia dovuto al fatto che il sistema eroga sempre lo stesso aroma. Il problema potrebbe essere facilmente risolto cambiando ogni giorno l'aroma utilizzato. Questa soluzione però presenta un inconveniente tecnico: cambiando aroma, nelle tubazioni si scatenano delle reazioni chimiche che producono esalazioni insopportabili. Per evitare questo problema, prima di cambiare aroma è necessario effettuare una pulizia delle tubazioni, mediante una sostanza che dipende dall'aroma da sostituire e dal nuovo aroma che si dovrà utilizzare.

Supponendo di avere  $n$  aromi differenti  $a_1, a_2, \dots, a_n$ , indichiamo con  $c_{i,j}$  il costo della pulizia quando, dopo l'aroma  $a_i$  si deve utilizzare l'aroma  $a_j$ ,  $i, j = 1, \dots, n$ . Si vuole trovare una permutazione  $(\pi_1, \pi_2, \dots, \pi_n)$  di  $(1, 2, \dots, n)$ , in modo che utilizzando ciclicamente gli aromi nell'ordine fornito dalla permutazione, cioè nell'ordine  $a_{\pi_1}, a_{\pi_2}, \dots, a_{\pi_n}$ , ricominciando da  $a_{\pi_1}$  dopo  $a_{\pi_n}$ , il costo totale per la pulizia sia minimo.

Nota: Chiamiamo *costo della permutazione*  $(\pi_1, \pi_2, \dots, \pi_n)$  la somma dei costi di pulizia per passare da un aroma all'altro tornando all'inizio, secondo l'ordine dato dalla permutazione. Formalmente può essere espresso come  $\sum_{i=1}^{n-1} c_{\pi_i, \pi_{i+1}} + c_{\pi_n, \pi_1}$ . Il problema è dunque quello di trovare una permutazione di costo minimo.

Esempio: Supponete  $n = 4$ , con i costi  $c_{i,j}$  dati dalla matrice riportata a destra. Il costo della permutazione  $(1, 3, 2, 4)$  è  $1 + 4 + 2 + 1 = 8$ , mentre il costo di  $(1, 4, 2, 3)$  è  $3 + 1 + 2 + 1 = 7$ . Si osservi che data una permutazione, facendo scorrere circolarmente gli elementi è possibile trovare una permutazione dello stesso costo che inizia da  $a_1$ . Ad esempio, la permutazione  $(4, 1, 3, 2)$  ha lo stesso costo di  $(1, 3, 2, 4)$ . Pertanto si possono considerare solo permutazioni  $(\pi_1, \pi_2, \dots, \pi_n)$  con  $\pi_1 = 1$ .

0	1	1	3
1	0	2	2
1	4	0	3
1	1	1	0

Cosa si richiede:

- Descrivete *sinteticamente a parole* e poi ad alto livello in pseudocodice, un algoritmo che utilizzando la *tecnica greedy* determini una permutazione cercando di minimizzare il costo.
- Fornite una stima dei tempi di calcolo dell'algoritmo ottenuto in funzione del numero  $n$  di aromi.
- Indicate se l'algoritmo trova sempre una soluzione ottima oppure no, motivando la risposta. In particolare, in caso di risposta negativa presentate un esempio in cui non viene trovato l'ottimo (indicate la matrice dei costi in ingresso, la soluzione ottenuta dall'algoritmo che avete scritto e una soluzione ottima, con i rispettivi costi).

Cognome.....

# Algoritmi e Strutture Dati

Nome.....

Prova scritta del 16 giugno 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 AVL e un albero 2-3 ottenuti inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri a partire da alberi inizialmente vuoti: 18 24 30 32 16 17 14

(a) Disegnate l'albero AVL	(b) Disegnate l'albero 2-3
(c) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ampiezza</i> dell'albero AVL	(f) L'albero AVL ottenuto al punto (a) è perfettamente bilanciato? <input type="checkbox"/>
(d) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ordine anticipato</i> dell'albero AVL	(g) Se avete risposto NO al punto (f), disegnate un albero <i>perfettamente bilanciato</i> contenente gli stessi numeri
(e) Scrivete l'elenco dei valori dei nodi ottenuto mediante la visita in <i>ordine posticipato</i> dell'albero AVL	

2. Considerate la seguente sequenza di numeri memorizzata in un array che deve essere ordinata in modo crescente: 444 212 323 515 432 616 232 818

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

3. Considerate funzione  $f : \{a, b, \dots, z\} \rightarrow \{0, 1, \dots, 15\}$  definita come segue:

$x$	$f(x)$
a	0
b	1
c	2
d	3
e	4
f	5
g	6
h	6
i	7

$x$	$f(x)$
j	7
k	7
l	8
m	9
n	10
o	10
p	11
q	12
r	12

$x$	$f(x)$
s	13
t	13
u	14
v	14
w	15
x	15
y	15
z	15

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Sia  $h$  la funzione che trasforma ogni parola  $k$  sull'alfabeto  $\{a, b, \dots, z\}$  nell'intero che si ottiene applicando  $f$  al primo carattere di  $k$  e  $g$  la funzione che trasforma ogni parola nel più piccolo numero primo maggiore o uguale al numero di vocali presenti in  $k$ .

Esempi:  $h(\text{gatto}) = 6$ ,  $g(\text{gatto}) = 2$ ,  $h(\text{lucertola}) = 8$ ,  $g(\text{lucertola}) = 5$ .

Inserite nella tabella hash a destra, inizialmente vuota, le seguenti parole, nell'ordine indicato:

leone cane formica iguana rinoceronte cornacchia  
dromedario bue

Come funzione hash utilizzate  $h$ . Per la gestione delle collisioni utilizzate l'hashing doppio mediante la funzione

$$c(k, i) = (h(k) + i \cdot g(k)) \bmod 16$$

4. L'eccentrico sindaco di una città di cui non si può riportare il nome, ha fatto costruire un sistema di aromaterapia che, utilizzando una rete di tubazioni, dovrebbe diffondere in tutte le abitazioni sostanze aromatiche gradevoli e stimolanti. Il sistema è stato messo in funzione da qualche settimana.

Dopo un breve periodo di grande euforia, gli abitanti della città hanno cominciato a mostrare forti segni di depressione e stanchezza. Gli psicologi che sono stati interpellati per studiare la situazione sono giunti alla conclusione che ciò sia dovuto al fatto che il sistema eroga sempre lo stesso aroma. Il problema potrebbe essere facilmente risolto cambiando ogni giorno l'aroma utilizzato. Questa soluzione però presenta un inconveniente tecnico: cambiando aroma, nelle tubazioni si scatenano delle reazioni chimiche che producono esalazioni insopportabili. Per evitare questo problema, prima di cambiare aroma è necessario effettuare una pulizia delle tubazioni, mediante una sostanza che dipende dall'aroma da sostituire e dal nuovo aroma che si dovrà utilizzare.

Supponendo di avere  $n$  aromi differenti  $a_1, a_2, \dots, a_n$ , indichiamo con  $c_{i,j}$  il costo della pulizia quando, dopo l'aroma  $a_i$  si deve utilizzare l'aroma  $a_j$ ,  $i, j = 1, \dots, n$ . Si vuole trovare una permutazione  $(\pi_1, \pi_2, \dots, \pi_n)$  di  $(1, 2, \dots, n)$ , in modo che utilizzando ciclicamente gli aromi nell'ordine fornito dalla permutazione, cioè nell'ordine  $a_{\pi_1}, a_{\pi_2}, \dots, a_{\pi_n}$ , ricominciando da  $a_{\pi_1}$  dopo  $a_{\pi_n}$ , il costo totale per la pulizia sia minimo.

Nota: Chiamiamo *costo della permutazione*  $(\pi_1, \pi_2, \dots, \pi_n)$  la somma dei costi di pulizia per passare da un aroma all'altro tornando all'inizio, secondo l'ordine dato dalla permutazione. Formalmente può essere espresso come  $\sum_{i=1}^{n-1} c_{\pi_i, \pi_{i+1}} + c_{\pi_n, \pi_1}$ . Il problema è dunque quello di trovare una permutazione di costo minimo.

Esempio: Supponete  $n = 4$ , con i costi  $c_{i,j}$  dati dalla matrice riportata a destra. Il costo della permutazione  $(1, 3, 2, 4)$  è  $1 + 4 + 2 + 1 = 8$ , mentre il costo di  $(1, 4, 2, 3)$  è  $3 + 1 + 2 + 1 = 7$ . Si osservi che data una permutazione, facendo scorrere circolarmente gli elementi è possibile trovare una permutazione dello stesso costo che inizia da  $a_1$ . Ad esempio, la permutazione  $(4, 1, 3, 2)$  ha lo stesso costo di  $(1, 3, 2, 4)$ . Pertanto si possono considerare solo permutazioni  $(\pi_1, \pi_2, \dots, \pi_n)$  con  $\pi_1 = 1$ .

$$\begin{pmatrix} 0 & 1 & 1 & 3 \\ 1 & 0 & 2 & 2 \\ 1 & 4 & 0 & 3 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Cosa si richiede:

- Descrivete *sinteticamente a parole* e poi ad alto livello in pseudocodice, un algoritmo che utilizzando la *tecnica greedy* determini una permutazione cercando di minimizzare il costo.
- Fornite una stima dei tempi di calcolo dell'algoritmo ottenuto in funzione del numero  $n$  di aromi.
- Indicate se l'algoritmo trova sempre una soluzione ottima oppure no, motivando la risposta. In particolare, in caso di risposta negativa presentate un esempio in cui non viene trovato l'ottimo (indicate la matrice dei costi in ingresso, la soluzione ottenuta dall'algoritmo che avete scritto e una soluzione ottima, con i rispettivi costi).