Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 13 luglio 2018

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.*

1. Considerate funzione $f : \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\} \to \{0, 1, \ldots, 15\}$ definita come segue:

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

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

k	$f(k)$
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 $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$ nell'intero che si ottiene applicando f al primo carattere di k e g la funzione che trasforma ogni parola k nel *più piccolo numero primo* maggiore o uguale alla lunghezza di k.

Esempi: $h(\mathtt{ratto}) = 12$, $g(\mathtt{ratto}) = 5$, $h(\mathtt{cane}) = 2$, $g(\mathtt{cane}) = 5$.

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

`tonno gallo serpe gallina toro gatto leone leopardo`

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

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

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:
`15 1 4 2 6 17 16 8`

(a) Supponete di ordinare la sequenza mediante l'algoritmo `heapSort`. Indicate il contenuto dell'array dopo averlo trasformato in uno heap.

(b) Supponete di ordinare la sequenza mediante l'algoritmo `insertionSort`. Indicate il contenuto dell'array *dopo la terza* iterazione del ciclo principale dell'algoritmo.

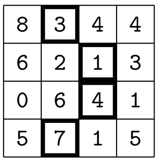
(c) Supponete di ordinare la sequenza mediante l'algoritmo `mergeSort`. Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive dell'algoritmo, prima del `merge` finale.

3. Data una matrice di interi, consideriamo cammini che contengono un numero per ciascuna riga e, per i numeri nelle righe successive alla prima, soddisfano i seguenti vincoli:

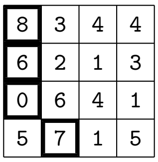
- un *numero pari* può seguire nel cammino *esclusivamente* l'elemento della riga precedente che si trova sulla stessa colonna;
- un *numero dispari* può seguire nel cammino l'elemento della riga precedente che si trova sulla colonna a sinistra (se esiste) oppure quello che si trova sulla colonna a destra (se esiste) dell'elemento considerato.

Il *peso* di un cammino è dato dalla somma dei numeri che lo costituiscono.

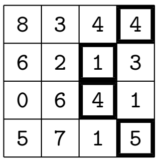
Esempio
Nelle figure seguenti sono evidenziati dei cammini in una matrice di pesi, rispettivamente, 15, 21, 14, 13 e 12.



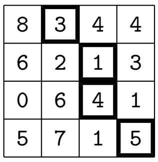
8	$\boxed{3}$	4	4
6	2	$\boxed{1}$	3
0	6	$\boxed{4}$	1
5	$\boxed{7}$	1	5



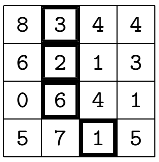
$\boxed{8}$	3	4	4
$\boxed{6}$	2	1	3
$\boxed{0}$	6	4	1
5	$\boxed{7}$	1	5



8	3	4	$\boxed{4}$
6	2	$\boxed{1}$	3
0	6	$\boxed{4}$	1
5	7	1	$\boxed{5}$



8	$\boxed{3}$	4	4
6	2	$\boxed{1}$	3
0	6	$\boxed{4}$	1
5	7	1	$\boxed{5}$



8	$\boxed{3}$	4	4
6	$\boxed{2}$	1	3
0	$\boxed{6}$	4	1
5	7	$\boxed{1}$	5

Cosa si richiede
Data una matrice M di dimensione $n \times n$, (dove con $m_{i,j}$ indichiamo l'elemento di riga i e colonna j) vogliamo determinare il minimo peso di un cammino ottenuto secondo le regole precedenti. Indichiamo con $c_{i,j}$ il minimo peso di un cammino che inizia in una qualunque posizione della prima riga e termina nella posizione (i, j). Si osservi che $c_{1,j} = m_{1,j}$, per $j = 1, \ldots, n$. Denotiamo con C la matrice dei valori $c_{i,j}$.

Esempio

$$\text{Data } M = \begin{bmatrix} 8 & 3 & 4 & 4 \\ 6 & 2 & 1 & 3 \\ 0 & 6 & 4 & 1 \\ 5 & 7 & 1 & 5 \end{bmatrix}, \text{ si ottiene } C = \begin{bmatrix} 8 & 3 & 4 & 4 \\ 14 & 5 & 4 & 7 \\ 14 & 11 & 8 & 5 \\ 16 & 15 & 6 & 13 \end{bmatrix}.$$

Dalla matrice C si deduce che il peso minimo di un cammino è 6.

Risolvete i seguenti punti:

(a) Tenendo conto delle regole con cui sono definiti i cammini, scrivete una o più formule o, in alternativa, righe di pseudocodice, che permettano di ricavare il generico valore $c_{i,j}$ della riga i di C, con $i > 1$, sulla base di alcuni valori della riga $i - 1$ di C e del valore $m_{i,j}$.

(b) Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo in ingresso la matrice M riempia la matrice C (utilizzate quanto scritto al punto (a)) e determini il peso minimo di un cammino.

(c) Fornite una stima del tempo totale utilizzato dall'algoritmo in funzione di n.

(d) Indicate come dalla matrice C sia possibile ricavare il cammino di peso minimo (se vi sono più cammini di peso minimo, è sufficiente ricavarne uno qualunque).

Note

- I punti precedenti ***devono essere risolti nell'ordine indicato.***
- ***Non cambiate i nomi*** delle matrici stabiliti nel testo dell'esercizio.

Cognome.....

Nome.....

Matricola.....

Algoritmi e Strutture Dati

Prova scritta del 13 luglio 2018

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.*

1. Considerate funzione $f : \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\} \to \{0, 1, \ldots, 15\}$ definita come segue:

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

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

k	$f(k)$
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 $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$ nell'intero che si ottiene applicando f al primo carattere di k e g la funzione che trasforma ogni parola k nel *più piccolo numero primo* maggiore o uguale alla lunghezza di k.

Esempi: $h(\mathtt{ratto}) = 12$, $g(\mathtt{ratto}) = 5$, $h(\mathtt{cane}) = 2$, $g(\mathtt{cane}) = 5$.

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

`tigre gatto talpa giraffa topo gallo lepre lombrico`

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

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

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:
`18 4 7 5 9 20 19 11`

(a) Supponete di ordinare la sequenza mediante l'algoritmo `heapSort`. Indicate il contenuto dell'array dopo averlo trasformato in uno heap.

(b) Supponete di ordinare la sequenza mediante l'algoritmo `insertionSort`. Indicate il contenuto dell'array *dopo la terza* iterazione del ciclo principale dell'algoritmo.

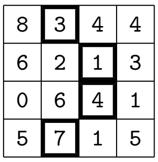
(c) Supponete di ordinare la sequenza mediante l'algoritmo `mergeSort`. Indicate il contenuto dell'array dopo l'esecuzione delle due chiamate ricorsive dell'algoritmo, prima del `merge` finale.

3. Data una matrice di interi, consideriamo cammini che contengono un numero per ciascuna riga e, per i numeri nelle righe successive alla prima, soddisfano i seguenti vincoli:

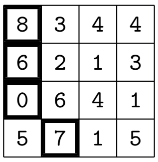
- un *numero pari* può seguire nel cammino *esclusivamente* l'elemento della riga precedente che si trova sulla stessa colonna;
- un *numero dispari* può seguire nel cammino l'elemento della riga precedente che si trova sulla colonna a sinistra (se esiste) oppure quello che si trova sulla colonna a destra (se esiste) dell'elemento considerato.

Il *peso* di un cammino è dato dalla somma dei numeri che lo costituiscono.

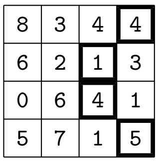
Esempio
Nelle figure seguenti sono evidenziati dei cammini in una matrice di pesi, rispettivamente, 15, 21, 14, 13 e 12.



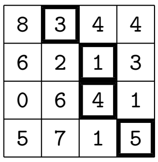
$$\begin{array}{|c|c|c|c|}\hline 8 & \boxed{3} & 4 & 4 \\ \hline 6 & 2 & \boxed{1} & 3 \\ \hline 0 & 6 & \boxed{4} & 1 \\ \hline 5 & \boxed{7} & 1 & 5 \\ \hline \end{array}$$



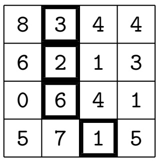
$$\begin{array}{|c|c|c|c|}\hline \boxed{8} & 3 & 4 & 4 \\ \hline \boxed{6} & 2 & 1 & 3 \\ \hline \boxed{0} & 6 & 4 & 1 \\ \hline 5 & \boxed{7} & 1 & 5 \\ \hline \end{array}$$



$$\begin{array}{|c|c|c|c|}\hline 8 & 3 & 4 & \boxed{4} \\ \hline 6 & 2 & \boxed{1} & 3 \\ \hline 0 & 6 & \boxed{4} & 1 \\ \hline 5 & 7 & 1 & \boxed{5} \\ \hline \end{array}$$



$$\begin{array}{|c|c|c|c|}\hline 8 & \boxed{3} & 4 & 4 \\ \hline 6 & 2 & \boxed{1} & 3 \\ \hline 0 & 6 & \boxed{4} & 1 \\ \hline 5 & 7 & 1 & \boxed{5} \\ \hline \end{array}$$



$$\begin{array}{|c|c|c|c|}\hline 8 & \boxed{3} & 4 & 4 \\ \hline 6 & \boxed{2} & 1 & 3 \\ \hline 0 & \boxed{6} & 4 & 1 \\ \hline 5 & 7 & \boxed{1} & 5 \\ \hline \end{array}$$

Cosa si richiede
Data una matrice M di dimensione $n \times n$, (dove con $m_{i,j}$ indichiamo l'elemento di riga i e colonna j) vogliamo determinare il minimo peso di un cammino ottenuto secondo le regole precedenti. Indichiamo con $c_{i,j}$ il minimo peso di un cammino che inizia in una qualunque posizione della prima riga e termina nella posizione (i,j). Si osservi che $c_{1,j} = m_{1,j}$, per $j = 1, \ldots, n$. Denotiamo con C la matrice dei valori $c_{i,j}$.

Esempio

$$\text{Data } M = \begin{bmatrix} 8 & 3 & 4 & 4 \\ 6 & 2 & 1 & 3 \\ 0 & 6 & 4 & 1 \\ 5 & 7 & 1 & 5 \end{bmatrix}, \text{ si ottiene } C = \begin{bmatrix} 8 & 3 & 4 & 4 \\ 14 & 5 & 4 & 7 \\ 14 & 11 & 8 & 5 \\ 16 & 15 & 6 & 13 \end{bmatrix}.$$

Dalla matrice C si deduce che il peso minimo di un cammino è 6.

Risolvete i seguenti punti:

(a) Tenendo conto delle regole con cui sono definiti i cammini, scrivete una o più formule o, in alternativa, righe di pseudocodice, che permettano di ricavare il generico valore $c_{i,j}$ della riga i di C, con $i > 1$, sulla base di alcuni valori della riga $i-1$ di C e del valore $m_{i,j}$.

(b) Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo in ingresso la matrice M riempia la matrice C (utilizzate quanto scritto al punto (a)) e determini il peso minimo di un cammino.

(c) Fornite una stima del tempo totale utilizzato dall'algoritmo in funzione di n.

(d) Indicate come dalla matrice C sia possibile ricavare il cammino di peso minimo (se vi sono più cammini di peso minimo, è sufficiente ricavarne uno qualunque).

Note

- I punti precedenti ***devono essere risolti nell'ordine indicato.***
- ***Non cambiate i nomi*** delle matrici stabiliti nel testo dell'esercizio.