

CORSO DI ALGORITMI E STRUTTURE DATI

Prof. ROBERTO PIETRANTUONO

Terza esercitazione

PROBLEMA.

Bisogna configurare una macchina per tagliare a pezzi delle travi di legno. I costi variano in base alla lunghezza della trave da tagliare. Si assume che la macchina faccia solo un taglio alla volta.

È facile notare che selezioni diverse nell'ordine di taglio possono portare a costi diversi. Ad esempio, si consideri una trave di 10 metri che deve essere tagliata a distanza 2, 4 e 7 metri da un'estremità. Ci sono diverse scelte. Si può tagliare prima a 2, poi a 4, poi a 7. Questo porta ad un costo di $10 + 8 + 6 = 24$ perché la prima trave era di 10 metri, la seconda, risultante dal primo taglio, di 8 e l'ultima, risultante dal secondo taglio, di 6. Un'altra scelta potrebbe essere tagliare a 4, poi a 2, poi a 7. Questo porterebbe a un costo di $10 + 4 + 6 = 20$, che è migliore.

Si progetti un algoritmo per determinare il costo minimo per tagliare una trave.

INPUT

L'input è costituito da diversi test. La prima riga di ogni test case conterrà un numero positivo l che rappresenta la lunghezza della trave da tagliare. Si assuma $l < 1000$. La riga successiva conterrà il numero n ($n < 50$) di tagli da eseguire.

La riga successiva è composta da n numeri positivi c_i ($0 < c_i < l$) che rappresentano i punti in cui devono essere eseguiti i tagli, dati in ordine strettamente crescente.

Un caso con $l = 0$ rappresenta la fine dell'input.

OUTPUT

Bisogna stampare il costo della soluzione ottima, cioè il costo minimo del taglio.

Sample Input

```
30
3
2 20 25
10
4
4 5 7 8
100
3
25 50 75
0
```

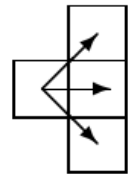
Sample Output

```
60
22
200
```

PROBLEMA

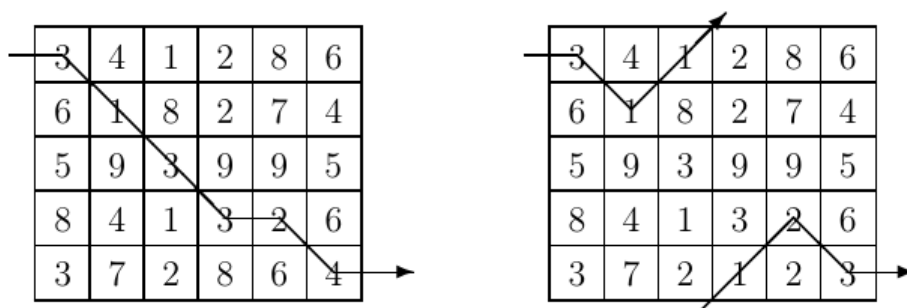
Bisogna trovare un percorso minimo attraverso una griglia di punti viaggiando solo da sinistra a destra.

Data una matrice di $m \times n$ pesi interi, scrivere un programma che calcoli un percorso di peso minimo. Un percorso inizia ad una qualsiasi riga nella colonna 1 (la prima colonna) e consiste in una sequenza di passaggi che termina nella colonna n (l'ultima colonna). Un passo consiste nel viaggiare dalla colonna i alla colonna $i + 1$ in una riga adiacente (orizzontale o diagonale). La prima e l'ultima riga (righe 1 e m) di una matrice sono considerate adiacenti, ovvero la matrice si "avvolge" in modo da rappresentare un cilindro orizzontale. I passi consentiti sono illustrati a destra.



Il peso totale di un percorso è la somma dei numeri interi in ciascuna delle n celle della matrice che vengono visitate.

Ad esempio, di seguito sono mostrate due matrici 5×6 leggermente diverse (l'unica differenza sono i numeri nella riga inferiore).



Il percorso minimo è illustrato per ogni matrice. Si noti che il percorso per la matrice a destra sfrutta la proprietà di adiacenza della prima e dell'ultima riga.

INPUT

L'input consiste in una o più sequenze che descrivono le matrici. La descrizione di una singola matrice è costituita dalle dimensioni di riga e colonna sulla prima riga, seguita da $m \times n$ interi dove m è la dimensione della riga e n è la dimensione della colonna. Gli interi compaiono nell'input in ordine di riga, ossia i primi n interi costituiscono la prima riga della matrice, i secondi n interi costituiscono la seconda riga e così via. Gli interi su una riga sono separati dagli altri interi da uno spazio.

Nota: i numeri interi devono essere positivi.

Ci saranno una o più specifiche di matrice in un file di input. L'input termina alla fine del file.

Per ogni specifica il numero di righe deve essere compreso tra 1 e 10 inclusi; il numero di colonne deve essere compreso tra 1 e 100 inclusi.

OUTPUT

L'output prevede due righe per ogni matrice; la prima riga riporta il percorso di peso minimo e la seconda riga il costo del percorso minimo. Il percorso di peso minimo è stampato come una sequenza di n interi (separati da uno o più spazi) che rappresentano le righe che costituiscono il percorso minimo. Se è presente più di un percorso di peso minimo, si stampi il percorso lessicograficamente più piccolo.

Nota: *Lessicograficamente* vuol dire l'ordine sulle successioni indotto dall'ordine sui loro elementi (es.: 1 4 5 7 è lessicograficamente minore di 1 5 3 9, perché il primo carattere è uguale, il secondo carattere della prima stringa è minore del secondo carattere della seconda stringa).

Sample Input

5 6
3 4 1 2 8 6
6 1 8 2 7 4
5 9 3 9 9 5
8 4 1 3 2 6
3 7 2 8 6 4
5 6
3 4 1 2 8 6
6 1 8 2 7 4
5 9 3 9 9 5
8 4 1 3 2 6
3 7 2 1 2 3
2 2
9 10
9 10

Sample Output

1 2 3 4 4 5
16
1 2 1 5 4 5
11
1 1
19