

# 1. Box Car Race (race)

## Descrizione del problema

Pierino si è iscritto alla *Box Car Race* delle montagne rocciose: una competizione in cui ogni concorrente si lancia giù dal cocuzzolo di una montagna in un veicolo senza motore autocostruito, e deve raggiungere l'arrivo sano e salvo con il veicolo ancora intatto. I concorrenti possono seguire il percorso che più gli aggrada, anche se i veicoli senza motore (anche piuttosto primitivi) sono in grado di muoversi solamente in discesa.

Pierino vuole vincere la gara di quest'anno, e quindi si è procurato una mappa di tutte le strade e sentieri della montagna su cui si svolge la gara. Dopo essere andato a fare un accurato sopralluogo, si è anche annotato per ogni strada e sentiero un coefficiente di *pericolosità*, cioè una stima del rischio di incidente fatale se il percorso della Box Car passa da quella strada.

Ora Pierino vuole scegliere un percorso da seguire poi durante la gara. Sapendo che, dato un qualunque percorso, la sua *pericolosità totale* è data dalla *massima* pericolosità delle strade da cui è composto, trova il percorso dalla partenza all'arrivo meno pericoloso!

## Dati di input

La prima riga del file `input.txt` contiene due numeri interi  $N$ ,  $M$ , rispettivamente il numero di incroci e il numero di strade tra questi incroci. L'incrocio 1 rappresenta la partenza, l'incrocio  $N$  rappresenta l'arrivo. Le successive  $M$  righe contengono ciascuna tre interi,  $x_i$ ,  $y_i$ ,  $p_i$ , che indicano la presenza di una strada in discesa tra l'incrocio  $x_i$  e l'incrocio  $y_i$  con pericolosità  $p_i$ .

## Dati di output

La prima e unica riga del file `output.txt` conterrà un unico numero intero  $P$ , la minima pericolosità dei percorsi tra la partenza e l'arrivo.

## Assunzioni

- $1 \leq N \leq 10000$
- $0 \leq M \leq 1000000$
- $1 \leq p_i \leq 1000000$
- Ogni incrocio è collegato con al più 1000 altri;
- Esiste almeno un percorso in discesa tra la partenza e l'arrivo.

## Esempi di input/output

File input.txt	File output.txt
<pre>6 7 1 2 1 1 3 8 2 3 5 3 4 3 4 6 9 4 5 7 5 6 2</pre>	<pre>7</pre>

## 2. Tre giorni in Tibet (tibet)

### Descrizione del problema

Vi aspetta una magnifica vacanza in ritiro nel monastero di Whou Tzun! Per raggiungere il monastero, tuttavia, dovete ancora attraversare l'impervia vallata di Shi-Ho. La vallata ha forma rettangolare ed è suddivisa in  $N \times M$  cocuzzoli, voi siete nell'angolo in alto a destra  $(1, 1)$  e il monastero di Whou Tzun è nell'angolo opposto  $(N, M)$ .

Fortunatamente avete a disposizione una mappa molto accurata della vallata, che riporta per ogni cocuzzolo l'altitudine corrispondente  $H_{ij}$ . Nel cercare di raggiungere il monastero, potete ogni volta muovervi di un passo a Nord, a Sud, a Ovest, o a Est (ma non in diagonale), e l'attraversamento vi richiederà un'ora se il cocuzzolo su cui vi state muovendo è ad un'altitudine pari o inferiore al cocuzzolo su cui eravate precedentemente, oppure due ore se il nuovo cocuzzolo è più in alto del vecchio.

Non vedete l'ora di raggiungere l'agognato monastero: quante ore, al minimo, vi richiederà il tragitto?

### Dati di input

La prima riga del file `input.txt` contiene due numeri interi,  $N$  ed  $M$ .

Le successive  $N$  righe contengono ciascuna  $M$  interi  $H_{ij}$ , che indicano l'altitudine del corrispondente cocuzzolo.

### Dati di output

Il file `output.txt` consisterà di un'unica riga, contenente un unico numero intero  $T$ , il numero di ore che è necessario impiegare per raggiungere il monastero di Whou Tzun.

### Assunzioni

- $1 \leq N \leq 2000$
- $1 \leq M \leq 2000$
- $1 \leq H_{ij} \leq 1000000$

### Esempi di input/output

File <code>input.txt</code>	File <code>output.txt</code>
10 4 1 10 9 6 8 1 8 7 9 2 7 8 10 3 6 9 11 4 5 10 10 10 1 11 11 9 2 12 9 8 3 13 11 7 4 14 12 6 5 1	16

### 3. Torre di Ahinoi (**ahinoi**)

#### Descrizione del problema

Nel nuovo parco a tema Disnéworld vogliono installare una replica della magnifica torre di Ahinoi. Per costruire questa torre hanno rilevato da un fallimento un lotto di lastre rettangolari di marmo, ciascuna dello stesso spessore ma con differente altezza e larghezza (tutti i valori di altezza e larghezza sono differenti tra loro).

Per costruire la torre è necessario che ogni lastra venga posata su una lastra più grande in entrambe le dimensioni, anche se ovviamente le lastre possono essere ruotate di 90 gradi se serve. Aiuta i proprietari di Disnéworld calcolando quanto alta al massimo può essere fatta la Torre di Ahinoi!

#### Dati di input

La prima riga del file `input.txt` contiene un unico numero intero,  $N$ , il numero di lastre.

Le successive  $N$  righe contengono ciascuna due interi,  $A_i$ ,  $B_i$ , che significano la presenza di una lastra di dimensioni  $A_i \times B_i$ .

#### Dati di output

Il file `output.txt` consisterà di un'unica riga, contenente un unico numero intero  $H$ , la massima altezza che può raggiungere una torre di Ahinoi con le lastre disponibili.

#### Assunzioni

- $1 \leq N \leq 1000$
- $1 \leq A_i, B_i \leq 1000000$

#### Esempi di input/output

File input.txt	File output.txt
4 7 4 2 3 1 1 6 5	3

File input.txt	File output.txt
5 6 4 7 5 8 1 9 2 10 3	3