

# Viaggio in treno (finestrini)

## Descrizione del problema

Carmen sta andando a Biella per preparare le OII! Purtroppo, nella carrozza del treno su cui viaggia, non funziona l'aria condizionata. Aiutala a decidere quali finestrini aprire.



Figure 1: “Il treno su cui viaggia Carmen”

La carrozza ha  $N$  file di posti, ciascuna con un finestrino sul lato destro e uno sul lato sinistro. Inizialmente i finestrini sono tutti chiusi. Ogni finestrino richiede una certa energia per essere aperto:  $L_i$  per il finestrino sinistro e  $R_i$  per il finestrino destro della fila  $i$ . Spendendo meno energia possibile, Carmen vuole aprire alcuni finestrini in modo che:

- in ciascuna fila, ci sia *esattamente* un finestrino aperto, a sinistra o a destra (non entrambi, per evitare correnti d'aria),
- non ci siano tre file consecutive che, su uno stesso lato, abbiano i finestrini tutti chiusi.

Stabilisci il minimo valore di energia necessaria, in totale, per aprire i finestrini in modo da soddisfare queste condizioni.

## Dati di input

La prima riga del file di input contiene un intero  $T$ , il numero di casi di test. Seguono  $T$  casi di test, numerati da 1 a  $T$ . Ogni caso di test è preceduto da una riga vuota.

Ogni caso di test è composto da  $N + 1$  righe:

- la prima riga contiene l'intero  $N$ ;
- le successive  $N$  righe contengono ciascuna i due interi  $L_i$  e  $R_i$ .

## Dati di output

Il file di output deve contenere la risposta ai casi di test che sei riuscito a risolvere. Per ogni caso di test che hai risolto, il file di output deve contenere una riga con la dicitura:

Case #t: e

dove t è il numero del caso di test (a partire da 1) e il valore e il minimo valore di energia necessario in totale.

## Assunzioni

- $T = 24$ , nei file di input che scaricherai saranno presenti esattamente 24 casi di test.
- $1 \leq N \leq 1000$ .
- $1 \leq L_i, R_i \leq 1\,000\,000$  per ogni  $0 \leq i \leq N - 1$ .

## Esempi di input/output

---

**Input:**

```
2
5
1 4
1 4
4 5
1 4
3 2
10
3 10
2 5
2 8
8 8
7 4
2 8
1 7
7 10
1 8
5 4
```

---

**Output:**

```
Case #1: 10
Case #2: 40
```

---

## Spiegazione

Nel **primo caso d'esempio** la carrozza ha  $N = 5$  file. Spendendo energia 10, possiamo aprire i seguenti finestrini:

- nella fila 0, a sinistra (spendendo energia 1)
- nella fila 1, a sinistra (spendendo energia 1)
- nella fila 2, a destra (spendendo energia 5)

- nella fila 3, a sinistra (spendendo energia 1)
- nella fila 4, a destra (spendendo energia 2)

Non è possibile soddisfare le condizioni spendendo meno di 10. Pertanto, il minimo valore di energia necessaria per soddisfare le condizioni è 10

Nel **secondo caso d'esempio** la carrozza ha  $N = 10$  file. Il minimo valore di energia necessaria per soddisfare le condizioni è 40.

# Ruota della fortuna (fortuna)

## Descrizione del problema

Giorgio gestisce una ruota della fortuna a premi.



Figure 1: “La ruota della fortuna di Giorgio”

La ruota è suddivisa in  $N$  spicchi uguali, numerati da 0 a  $N - 1$ , in senso orario. Lo spicchio  $i$  corrisponde a un premio di un certo valore  $V_i$ .

I premi vengono assegnati in questo modo. Dopo aver comprato il biglietto, ognuno dei giocatori si posiziona in corrispondenza di uno spicchio a sua scelta, e non si muove più. Quando tutti i giocatori sono in posizione, la ruota viene fatta girare. Quando finalmente la ruota si ferma, ciascun giocatore vince un premio in base al valore del nuovo spicchio che si trova davanti a lui.

Giorgio ha taroccato il meccanismo della ruota per fermarla quando gli conviene di più. Ti viene dato il valore dei premi per ogni spicchio, e il numero  $G_i$  di giocatori posizionati davanti a ciascuno spicchio  $i$ . Aiuta Giorgio a calcolare la somma totale dei premi che dovrà pagare, fermando la ruota in modo che questa somma sia minima!

## Dati di input

La prima riga del file di input contiene un intero  $T$ , il numero di casi di test. Seguono  $T$  casi di test, numerati da 1 a  $T$ . Ogni caso di test è preceduto da una riga vuota.

Ogni caso di test è composto da 3 righe:

- la prima riga contiene l'intero  $N$ ;
- la seconda riga contiene il valore dei premi  $V_0, \dots, V_{N-1}$  di ciascuno spicchio;
- la terza riga contiene il numero di giocatori  $G_0, \dots, G_{N-1}$  che si sono posizionati in corrispondenza di ciascuno spicchio.

## Dati di output

Il file di output deve contenere la risposta ai casi di test che sei riuscito a risolvere. Per ogni caso di test che hai risolto, il file di output deve contenere una riga con la dicitura:

Case #t: x

dove t è il numero del caso di test (a partire da 1) e il valore x è somma totale dei premi minima.

## Assunzioni

- $T = 13$ , nei file di input che scaricherai saranno presenti esattamente 13 casi di test.
- $2 \leq N \leq 1000$ .
- $0 \leq V_i \leq 1000$  per  $i = 0, \dots, N - 1$ .
- $0 \leq G_i \leq 1000$  per  $i = 0, \dots, N - 1$ .

## Note

- La ruota può solo fermarsi ruotata esattamente di un *numero intero di spicchi* rispetto alla posizione iniziale.
- Se due o più giocatori si posizionano sullo stesso spicchio, NON si dividono il premio. Bensì, ciascuno di loro vince un premio pari all'intero valore dello spicchio. ## Esempi di input/output

---

### Input:

```
3
5
7 2 5 3 4
1 0 0 1 0
5
7 2 5 3 4
2 4 1 3 7
5
7 2 5 3 4
3 8 1 3 0
```

---

### Output:

```
Case #1: 5
Case #2: 61
Case #3: 51
```

---

## Spiegazione

In tutti e tre i casi la ruota ha  $N = 5$  spicchi, con premi di valore 7 euro, 2 euro, 5 euro, 3 euro e 4 euro.

Nel **primo caso d'esempio**, ci sono solo due giocatori. La somma minima dei premi è 5 euro, e si ottiene fermando la ruota ruotata di 2 spicchi in senso orario. In questo modo il primo giocatore riceverà un premio di 3 euro e il secondo un premio di 2 euro.

Nel **secondo caso d'esempio**, la somma minima dei premi è 61 euro, e si ottiene fermando la ruota ruotata di 3 spicchi in senso orario. (Quindi 2 giocatori riceveranno un premio da 5 euro, 4 da 3 euro, 1 da 4 euro, 3 da 7 euro, e 7 da 2 euro. La somma totale dei premi in euro è data da  $2 \cdot 5 + 4 \cdot 3 + 1 \cdot 4 + 3 \cdot 7 + 7 \cdot 2 = 61$ .)

Nel **terzo caso d'esempio**, la somma minima dei premi è 51 euro, e si ottiene fermando la ruota esattamente nella posizione di partenza.

# Gruppo vacanze (vacanze)

## Descrizione del problema

Alessandro ha vinto alla lotteria un pacchetto *all-inclusive* per una vacanza a Budapest per quattro persone! Sfortunatamente, i suoi numerosi impegni non gli permettono di andare in vacanza. Per questo ha deciso di individuare, fra i suoi conoscenti, un gruppo di quattro persone a cui regalare la vacanza.



Figure 1: “Veduta aerea di Budapest con il Danubio”

Alessandro ha  $N$  conoscenti, numerati da 0 a  $N - 1$ . Alcuni di essi sono amici fra loro. In particolare, fra i conoscenti di Alessandro esistono  $M$  coppie di amici (l'amicizia è sempre reciproca: se  $A$  è amico di  $B$ , anche  $B$  è amico di  $A$  e viceversa).

Secondo Alessandro, un gruppo di 4 persone è un *gruppo vacanza perfetto* se ognuno di loro ha già due amici nel gruppo, ma anche una persona nuova con cui fare amicizia (in altre parole, se ciascuno dei quattro membri è amico di **esattamente due** degli altri tre).

Aiutalo a contare quanti sono tutti i *gruppi vacanza perfetti* possibili fra i suoi conoscenti.

## Dati di input

La prima riga del file di input contiene un intero  $T$ , il numero di casi di test. Seguono  $T$  casi di test, numerati da 1 a  $T$ . Ogni caso di test è preceduto da una riga vuota.

Ogni caso di test è composto da  $M + 1$  righe:

- la prima riga contiene gli interi  $N$  e  $M$ ;
- la  $i$ -esima ( $0 \leq i < M$ ) delle successive  $M$  righe contiene due interi  $A_i$  e  $B_i$ , a significare che le persone  $A_i$  e  $B_i$  sono amici.

## Dati di output

Il file di output deve contenere la risposta ai casi di test che sei riuscito a risolvere. Per ogni caso di test che hai risolto, il file di output deve contenere una riga con la dicitura:

Case #t: g

dove t è il numero del caso di test (a partire da 1) e il valore g è il numero di gruppi vacanza perfetti possibili.

## Assunzioni

- $T = 19$ , nei file di input che scaricherai saranno presenti esattamente 19 casi di test.
- $4 \leq N \leq 1500$ .
- $0 \leq M \leq 3000$ .
- $0 \leq A_i, B_i < N$  per  $i = 0, \dots, M - 1$ .
- $A_i \neq B_i$  per  $i = 0, \dots, M - 1$  (nessuna persona è amica di se stessa).
- Le  $M$  coppie di amici sono tutte distinte.

Nei primi 8 casi di test valgono le seguenti assunzioni aggiuntive:

- $N \leq 50$ .
- $M \leq 1000$ .

## Note

- Due gruppi vacanza contano come diversi se c'è almeno una persona che appartiene al primo gruppo ma non al secondo. Cioè, l'ordine delle persone all'interno di un gruppo non è importante.

## Esempi di input/output

---

### Input:

```
3
5 6
4 0
3 4
1 2
1 4
3 2
0 2
4 6
0 1
0 2
0 3
1 2
1 3
2 3
7 15
2 3
6 0
3 4
2 6
3 5
```

5 2  
6 4  
2 0  
2 1  
4 1  
4 5  
0 5  
6 1  
5 6  
3 0

---

**Output:**

Case #1: 3  
Case #2: 0  
Case #3: 4

---

**Spiegazione**

Nel **primo caso d'esempio**, le  $M = 6$  relazioni di amicizia tra gli  $N = 5$  conoscenti di Alessandro sono mostrate in figura (a), dove le persone sono indicate da un cerchio e le amicizie da linee.

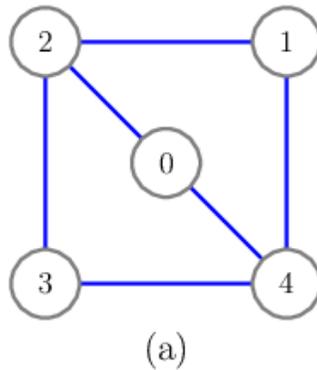
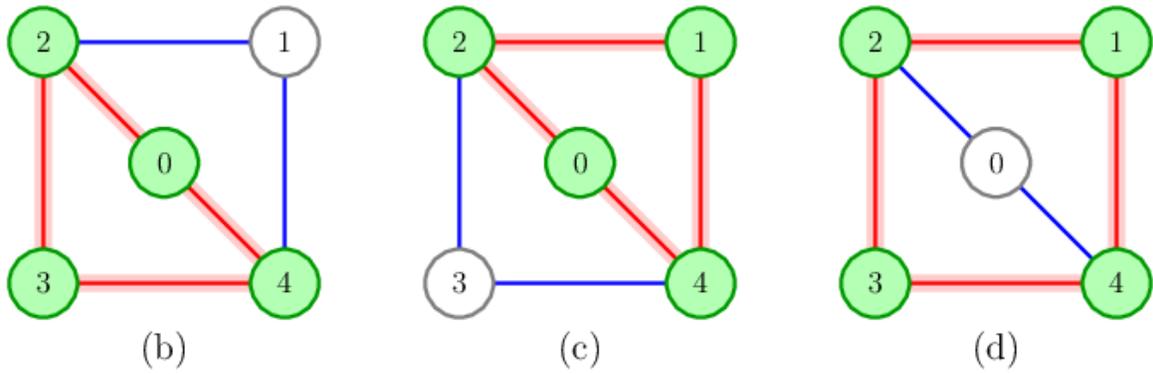


Figure 2: "Figura 1a"

Ci sono 3 possibili gruppi vacanza perfetti:  $\{0, 2, 3, 4\}$ ,  $\{0, 1, 2, 4\}$  e  $\{1, 2, 3, 4\}$ , mostrati rispettivamente nelle figure (b), (c) e (d).



Nel **secondo caso d'esempio**, ci sono  $N = 4$  persone e sono tutte amiche tra loro. Pertanto, è impossibile selezionare un gruppo di quattro persone come richiesto, perché ciascuno sarebbe amico di tutti e tre gli altri membri del gruppo.

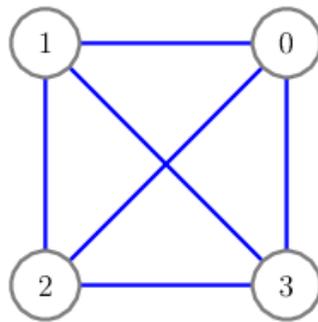


Figure 3: "Figura 2"

Nel **terzo caso d'esempio**, ci sono  $N = 7$  persone e  $M = 15$  relazioni di amicizia. In totale ci sono 4 possibili gruppi vacanza perfetti.

# Antichi fossili (fossili)

## Descrizione del problema

Massimo e i suoi amici sono alla ricerca di fossili antichi nella sterminata Death Valley.



Figure 1: “Uno dei fossili rinvenuti da Massimo.”

Massimo ha trovato un fossile con ben tre specie diverse di animali ormai estinti, indicati con  $a$ ,  $b$  e  $c$ . Per ciascuno di questi animali, ha cercato in rete l’anno di comparsa sulla Terra e l’anno di estinzione. Ecco cosa ha trovato.

L’animale  $a$  è comparso per la prima volta sulla Terra nell’anno  $a_1$  (misurato a partire dalla nascita della Terra), e si è estinto nell’anno  $a_2$  (cioè,  $a_2$  è il primo anno in cui l’animale non era più presente sulla Terra). Allo stesso modo, gli animali  $b$  e  $c$  sono comparsi per la prima volta sulla Terra rispettivamente negli anni  $b_1$  e  $c_1$ , e si sono estinti negli anni  $b_2$  e  $c_2$ .

Il fossile, per poter contenere tutte e tre le specie  $a$ ,  $b$  e  $c$ , deve risalire a un periodo in cui erano tutte presenti sulla Terra allo stesso tempo. Basandoti su queste informazioni, aiuta Massimo a capire quanto è lungo il periodo a cui potrebbe risalire il fossile.

## Dati di input

La prima riga del file di input contiene un intero  $T$ , il numero di casi di test. Seguono  $T$  casi di test, numerati da 1 a  $T$ . Ogni caso di test è preceduto da una riga vuota.

Ogni caso di test è composto da tre righe, contenenti rispettivamente gli interi

- $a_1$  e  $a_2$ ,
- $b_1$  e  $b_2$ ,
- $c_1$  e  $c_2$ .

## Dati di output

Il file di output deve contenere la risposta ai casi di test che sei riuscito a risolvere. Per ogni caso di test che hai risolto, il file di output deve contenere una riga con la dicitura:

Case #t: n

dove t è il numero del caso di test (a partire da 1) e il valore n è il numero di anni diversi in cui tutte e tre le specie a, b e c sono presenti sulla Terra.

## Assunzioni

- $T = 8$ , nei file di input che scaricherai saranno presenti esattamente 8 casi di test.
- Tutti gli animali sono comparsi non prima dell'anno zero e si sono estinti entro il miliardesimo anno. Quindi:
  - $0 \leq a_1 < a_2 \leq 10^9$ ,
  - $0 \leq b_1 < b_2 \leq 10^9$ ,
  - $0 \leq c_1 < c_2 \leq 10^9$ .
- È garantito che tutte e tre le specie erano presenti allo stesso tempo sulla Terra per almeno un intero anno.

## Esempi di input/output

---

**Input:**

2

15 43

20 500

7 30

70 100

70 100

70 100

---

**Output:**

Case #1: 10

Case #2: 30

---

## Spiegazione

Nel **primo caso d'esempio** ci sono 10 possibili anni a cui il fossile potrebbe risalire, gli anni sono: 20, 21, 22, ... fino a 29 compreso.

Nel **secondo caso d'esempio** ci sono 30 possibili anni a cui il fossile potrebbe risalire: 70, 71, 72, ... fino a 99 compreso.