

Caccia agli interruttori (interruttori [102 punti])

Questo problema è preso dalla fase territoriale delle Oii (Olimpiadi Italiane di Informatica).



Dato un set $L = \{0, 1, \dots, n - 1\}$ di n lampadine, si cui inizialmente solo la lampadina i è accesa, vuoi raggiungere lo stato in cui ogni lampadina è spenta agendo sul minor numero di interruttori possibile. Consideriamo questo numero $C(i)$ come il *costo energetico della lampadina i* . Indichiamo con S il sottoinsieme di quelle lampadine di L che dispongono di un proprio interruttore privato; chiaramente, se $i \in S$ allora $C(i) = 1$. Sono inoltre presenti degli interruttori associati a coppie di lampadine. La regola generale è che quando si agisce su un interruttore si inverte lo stato acceso/spento di tutte le lampadine associate ad esso.

Con quanta efficienza sai rispondere alle seguenti domande?

[C0]: quanto vale $C(0)$?

[min_sol_0]: su quali $C(0)$ interruttori agire per spegnere la lampadina 0?

[Cmax]: su quanti interruttori si dovrà agire nel caso peggiore? Individua un $i \in L$ che massimizzi $C(i)$ e ritorna sia un tale i che $C(i)$.

[min_sol_max]: su quali $C(i)$ interruttori agire per spegnere la lampadina i ritornata per il goal precedente?

Uno spunto enigmatico: [quantificatori in gioco][Quando si vogliono fare più cose al costo di una può essere una buona idea rovesciare qualcosa.]

Input

Si legga l'input da stdin. La prima riga contiene T , il numero di testcase (istanze) da risolvere. Seguono T istanze del problema, dove ogni istanza è descritta in $t_1 + t_2 + 1$ righe: la prima contiene i tre numeri $n, t_1, t_2 \in \mathbb{N}$ separati da uno spazio: il numero di lampadine (n), il numero di interruttori di tipo 1 (in S) e il numero di interruttori di tipo 2 (associati a 2 lampadine). Ciascuna delle seguenti t_1 righe

è composta da un unico intero, e specifica una lampadina su cui agisce un interruttore di tipo 1. Le successive t_2 righe contengono ciascuna una coppia di interi separati da spazio e rappresentano un interruttore di tipo 2.

Output

Per ciascuna istanza, prima di leggere l'istanza successiva, scrivi su `stdout` il tuo output così strutturato:

[**goal 1 (C0)**]: la prima riga contiene l'intero m , il minimo numero di interruttori su cui agire per accendere la sola lampadina 0.

[**goal 2 (min_sol_0)**]: le seguenti $C(0)$ righe riportano, uno per riga, $C(0)$ interruttori agendo sui quali si inverte lo stato della sola lampadina 0.

[**goal 3 (Cmax)**]: la terza riga riporta gli interi $\arg \max_i C(i)$ e $\max_i C(i)$ separati da spazio.

[**goal 4 (min_sol_max)**]: le seguenti $\max_i C(i)$ righe riportano, uno per riga, $\max_i C(i)$ interruttori agendo sui quali si riesce ad invertire lo stato della sola lampadina ritornata per il goal precedente.

Oltre alle dimensioni delle istanze, ogni subtask precisa quanti punti competono ai vari goal. Per ciascuna istanza di quel subtask si otterranno tutti i punti dei goal correttamente evasi (purché si rispetti almeno il formato in tutte le righe di output, incluse quelle che competono agli altri goal – altrimenti salta il protocollo di comunicazione tra il tuo programma risolutore e il server).

Assunzioni

1. $n \geq 1$
2. Tutti gli interruttori sono diversi.
3. È garantito che sia possibile spegnere ogni lampadina.

Esempio di Input/Output

<start in>	6 3 7	5 1 4			
4	0	4			5
4 1 4	4	0 1			4
1	1	1 2			4 3
0 1	0 1	2 3			3 2
0 2	3 2	3 4			2 1
1 2	1 3	2 2 1			1 0
3 2	0 3	0			0 5
<more>	3 4	1			4
	4 5	0 1			4 3
	2 5				3 2
	<more>	<end>			2 1
					1 0
					1
					0
					0 1
					0
					<end>

Spiegazione: Nel primo caso d'esempio l'impianto elettrico è il seguente:

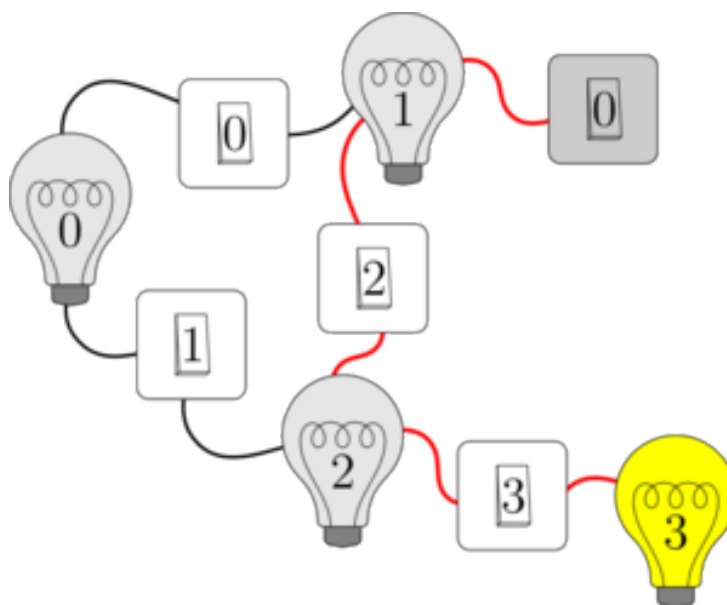


Figure 2: Configurazione iniziale dei computer e dei pulsanti.

Dovessi spegnere la sola lampadina 0 dovrei agire sull'interruttore privato della lampadina 1 (ossia l'interruttore t1 0) e sull'interruttore t2 0 che a quel punto spegnerebbe sia la lampadina 0 che la lampadina 1, portandomi nella situazione desiderata dove ogni lampadina è infine spenta. Per spegnere la sola lampadina 3 devo premere almeno 3 interruttori (caso peggiore):

1. premendo l'interruttore 3 di tipo 2 spengo la lampadina 3 e accendo la lampadina 2,
2. premendo l'interruttore 2 di tipo 2 spengo la lampadina 2 e accendo la lampadina 1,
3. premendo l'unico interruttore di tipo 1 spengo la lampadina 1.

Nel secondo caso d'esempio l'impianto elettrico è il seguente:

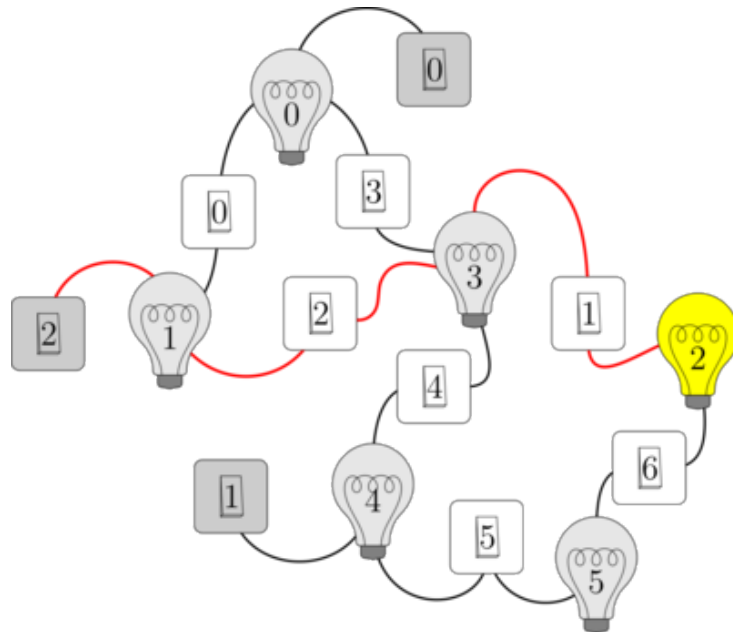


Figure 3: Configurazione iniziale dei computer e dei pulsanti.

Dovessi spegnere la sola lampadina 0 mi basterebbe agire sul suo interruttore privato (ossia l'interruttore t1 0). Per spegnere la lampadina 2 devo premere almeno 3 interruttori, ci sono più modi, uno è il seguente:

1. premendo l'interruttore 1 di tipo 2 spengo la lampadina 2 e accendo la lampadina 3,
2. premendo l'interruttore 2 di tipo 2 spengo la lampadina 3 e accendo la lampadina 1,
3. premendo l'interruttore 2 di tipo 1 spengo la lampadina 1.

Nel terzo caso d'esempio l'impianto elettrico è il seguente:

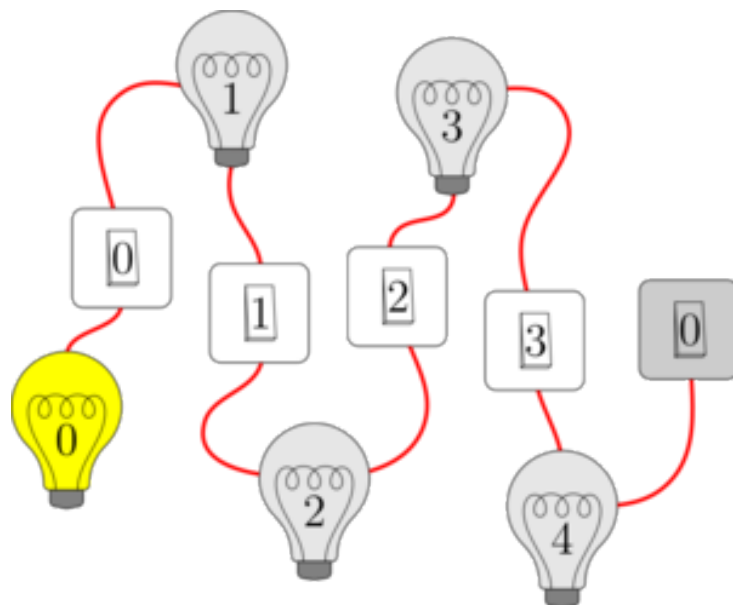


Figure 4: Configurazione iniziale dei computer e dei pulsanti.

Qui per spegnere la lampadina 0 bisogna premere tutti e 5 gli interruttori.

Nel quarto caso d'esempio l'impianto elettrico è il seguente:



Figure 5: Configurazione iniziale dei computer e dei pulsanti.

Sia la prima lampadina che la seconda si possono spegnere con soltanto un interruttore, pertanto entrambe le lampadine sono soluzioni valide come $\arg \max_i C(i)$ per il terzo goal.

Subtask

Il tempo limite per istanza (ossia per ciascun testcase) è sempre di 1 secondo.

I testcase sono raggruppati nei seguenti subtask.

1. [0 pts ← 4 istanze da 0 + 0 + 0 + 0 + 0 punti] **esempi_testo**: i quattro esempi del testo
2. [20 pts ← 4 istanze da 1 + 1 + 1 + 1 + 1 punti] **tiny**: $n = 10, t_1 = 3, t_2 = 50$
3. [20 pts ← 4 istanze da 1 + 1 + 1 + 1 + 1 punti] **small**: $n \leq 20, t_1 \leq 7, t_2 \leq 50$
4. [12 pts ← 4 istanze da 1 + 0 + 1 + 1 + 0 punti] **medium**: $n \leq 100, t_1 \leq 20, t_2 \leq 200$
5. [20 pts ← 10 istanze da 1 + 0 + 1 + 0 + 0 punti] **big**: $n \leq 20,000, t_1 \leq 1000, t_2 \leq 100,000$
6. [30 pts ← 15 istanze da 1 + 0 + 1 + 0 + 0 punti] **large**: $n \leq 50,000, t_1 \leq 2000, t_2 \leq 200,000$

In generale, quando si richiede la valutazione di un subtask vengono valutati anche i subtask che li precedono, ma si evita di avventurarsi in subtask successivi fuori dalla portata del tuo programma che potrebbe andare in crash o comportare tempi lunghi per ottenere la valutazione completa della sottomissione. Ad esempio, chiamando^{1, 2}:

```
rtal -s <URL> connect -x <token> -a size=small
interruttori -- python my_solution.py
```

vengono valutati, nell'ordine, i subtask:

esempi_testo, tiny, small.

Il valore di default per l'argomento size è large che include tutti i testcase.

¹<URL> server esame: [wss://ta.di.univr.it/esame](https://ta.di.univr.it/esame)

²<URL> server esercitazioni e simula-prove: [wss://ta.di.univr.it/algo](https://ta.di.univr.it/algo)