

Insiemi massimali di nodi mutualmente incomunicanti (mutually_unreachable [50 punti])

In un grafo *non-orientato* $G = (V, E)$ vi è un cammino p che porta da un nodo u a un nodo v se e solo se vi è un cammino da v ad u (basta percorrere p a ritroso). Quando esiste, un tale cammino *connette* i nodi u e v . Su V , la relazione binaria *essere connessi* è una relazione di equivalenza (riflessiva, simmetrica e transitiva). Dato un grafo non-orientato $G = (V, E)$, il tuo obiettivo è individuare un sottoinsieme S di V tale che nessun cammino in G connetta due diversi nodi in S .

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 un diverso grafo $G = (V, E)$. Per ogni istanza, la prima riga contiene due numeri interi separati da uno spazio: il numero di nodi $n = |V|$, e il numero di archi $m = |E|$. Seguono m righe ciascuna delle quali riporta un diverso arco di G . Ciascun arco viene specificato fornendo i nomi dei due nodi che collega (due numeri interi nell'intervallo $[0, n - 1]$, separati da uno spazio).

Output

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

- + la prima riga contiene un numero intero s , la massima cardinalità di un insieme di nodi di G tale che nessun cammino in G connetta due diversi nodi in S .
- + la riga seguente contiene s numeri interi separati da spazio. Tali interi, tutti contenuti nell'intervallo $[0, n - 1]$, sono i nomi dei nodi contenuti in S .

Esempio

Input

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

Output

1
3
2
0 5

Spiegazione: il primo grafo è connesso e quindi non è possibile includere in S più di un singolo nodo (gli insiemi S ottimi sono quanti i nodi, si sarebbe potuto ritornare un qualsiasi altro nodo in V al posto del nodo di nome 3). Nel caso del secondo grafo è facile verificare che tra i nodi 0 e 5 non è presente alcun cammino. Al tempo stesso, comunque si scelgano tre nodi almeno due di essi saranno connessi da un qualche cammino. Era pertanto corretto rispondere con una qualsiasi delle $3 \times 4 = 12$ coppie di nodi incomunicanti.

Subtask

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

I testcase sono raggruppati nei seguenti subtask.

1. [2 pts← 2 istanze da 1 punto] **esempi_testo:** i due esempi del testo
2. [12 pts←12 istanze da 1 punto] **small:** $N \leq 10, M \leq 20$
3. [18 pts←18 istanze da 1 punto] **medium:** $N \leq 100, M \leq 500$
4. [18 pts←18 istanze da 1 punto] **big:** $N \leq 5,000, M \leq 20,000$

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=medium  
mutually_unreachable -- python my_solution.py
```

vengono valutati, nell'ordine, i subtask:

esempi_testo, small, medium.

Il valore di default per l'argomento size è big 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)