pratica con la visita BFS (`BFS` [150 punti])

Sia $G = (V, E)$ un grafo diretto, con $V = \{0, ..., n-1\}$ e $E \subseteq \{(u, v) : u, v \in V, u \neq v\}$. Assumiamo quindi che E non contenga cappi (archi della forma (v, v)) nè archi paralleli (due archi $e_1 = (u_1, v_1)$ e $e_2 = (u_2, v_2)$ si dicono paralleli se $u_1 = u_2$ e $v_1 = v_2$). E' garantito che per ogni v in V esista in G un cammino dal nodo 0 al nodo v ed indichiamo con $d_G(0, v)$ il minimo numero di archi per un tale cammino (quindi, $d(0, 0) = 0$ e quando non serve precisare il pedice G scriviamo d invece che d_G). La prima sfida consiste nel calcolare la distanza $d(0, v)$ per ogni nodo $v \in V$. Come seconda sfida ti chiediamo di tornare un albero BFS dal nodo 0, ossia un sottografo $T = (V, E')$ di $G = (V, E)$ con $E' \subseteq E$ e $|E'| = n - 1$ per il quale $d_T(0, v) = d_G(0, v)$ per ogni nodo $v \in V$. La combinazione di queste condizioni implica che per ogni nodo $v \in V - \{0\}$ ci sia uno ed un solo nodo $u \in V$ tale che $(u, v) \in E'$; tale nodo è detto il padre di v in T ed è indicato con $\text{dad}_T(v)$ (per convenzione, il padre di $v = 0$ è definito come $\text{dad}_T(0) = 0$ benchè T non abbia alcun arco che entra nel nodo 0). Come ulteriore richiesta, computa il numero di alberi BFS (modulo 1.000.000.007).

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 spazi: 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 $(u, v) \in E$ va specificato fornendo nell'ordine u e v (due numeri interi nell'intervallo $[0, n-1]$, separati da spazi).

Output

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

[**goal 1**]: la prima riga su `stdout` contiene le distanze $d(0, 0), d(0, 1), ..., d(0, n-1)$ in questo ordine e separate da spazi.

[**goal 2**]: la seconda riga su `stdout` offre un BFS-tree $T = (V, E')$ di G specificando il padre di ogni nodo. In pratica contiene gli n numeri $\text{dad}_T(0), \text{dad}_T(1), ..., \text{dad}_T(n-1)$ in questo ordine e separati da spazi.

[**goal 3**]: la terza riga su `stdout` contiene il resto della divisione che ha il numero di BFS-trees come dividendo e $1.000.000.007$ come divisore.

Oltre alle dimensioni delle istanze, ogni subtask precisa quanti punti competono ai vari goal. Il punteggio ottenuto per la generica istanza di quel subtask sarà la somma dei punti per i goal dove la risposta è corretta (ma è necessario che sulle righe di output che competono agli altri goal sia quantomeno corretto il formato, per non far saltare il protocollo di comunicazione tra il tuo programma risolutore e il server).

Esempio di Input/Output

Input da `stdin`

```
-start-
2
4  6
0 2
0 3
2 1
3 1
1 0
1 2
-more-
```

```
6  9
0 2
0 4
1 0
2 3
2 5
3 1
4 2
4 5
5 1
 -end-
```

Output su `stdout`

```
0 2 1 1
0 2 0 0
2
0 3 1 2 1 2
0 3 0 2 0 2
4
```

Subtask

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

I testcase sono raggruppati nei seguenti subtask.

1. **[6 pts← 2 istanze da 1 + 1 + 1 punti] `esempi_testo`:** i due esempi del testo
2. **[36 pts←12 istanze da 1 + 1 + 1 punti] `small`:** $n \leq 10, m \leq 20$
3. **[54 pts←18 istanze da 1 + 1 + 1 punti] `medium`:** $n \leq 100, m \leq 500$
4. **[54 pts←18 istanze da 1 + 1 + 1 punti] `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
         BFS  -- 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`

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