

## Tropical Garden

El profesor de botánica Somhed, llevará varios grupos de estudiantes a uno de los jardines tropicales más grandes de Tailandia. El jardín está formado por  $N$  fuentes (numeradas  $0, 1, \dots, N-1$ ) y  $M$  caminos. Cada camino conecta un par de fuentes y puede ser andado en ambos sentidos. Para cada fuente hay al menos un camino que llega a ella. Los caminos presentan bellas colecciones botánicas que a Somhed le gustaría ver. Cada grupo puede iniciar su paseo en cualquier fuente.

Somhed *adora* las plantas tropicales. Por lo que estando en una fuente él y el grupo con el que está tomarán siempre el camino más bello que salga de ella, *a menos* que el camino más bello sea por el que llegaron y exista otra alternativa. En ese caso, elegirán salir por el segundo camino más bello. Por supuesto, si no hay alternativa, regresarán de nuevo por el camino que llegaron. Nota que siendo Somhed un botanista profesional no hay dos caminos que sean igualmente bellos para él.

Sus estudiantes no están muy interesados en plantas. Sin embargo, quieren comer en un restaurante especial que se encuentra junto a la fuente número  $P$ . Somhed sabe que sus estudiantes querrán comer después de andar exactamente  $K$  caminos, donde  $K$  puede ser distinto para cada grupo de estudiantes. Somhed se pregunta cuántas rutas distintas puede escoger para cada grupo dado que:

- Cada grupo puede iniciar en cualquier fuente
- Los caminos a tomar deben seleccionarse de la manera descrita arriba y
- Cada grupo debe terminar en la fuente  $P$  tras haber recorrido exactamente  $K$  caminos.

Nota que también pudieron haber pasado por la fuente  $P$  en algún momento previo de su paseo.

### Tu tarea

Dada la configuración de las fuentes y caminos, deberás encontrar la respuesta para  $Q$  grupos de estudiantes, es decir, para  $Q$  valores distintos de  $K$ .

Implementa el procedimiento `count_routes(N,M,P,R,Q,G)` el cual recibe los parámetros:

- ⤴  $N$  – El número de fuentes. Las fuentes se numeran del  $0$  al  $N-1$ .
- ⤴  $M$  – El número de caminos. Los caminos se numeran de  $0$  a  $M-1$  y además en orden decreciente de belleza, es decir, para  $0 \leq i < M-1$ , el trayecto  $i$  es más bello que el trayecto  $i+1$ .
- ⤴  $P$  – La fuente en donde se encuentra el restaurant especial.
- ⤴  $R$  – Un arreglo bidimensional que representa los caminos. Para  $0 \leq i < M$ , el camino  $i$  conecta las fuentes  $R[i][0]$  y  $R[i][1]$ . Recuerda que cada camino conecta dos fuentes distintas y además no hay dos caminos que conecten el mismo par de fuentes.
- ⤴  $Q$  – El número de grupos de estudiantes.
- ⤴  $G$  – Un arreglo unidimensional de enteros conteniendo los valores  $K$ . Para  $0 \leq i < Q$ , el valor  $G[i]$  representa el número de caminos  $K$  que el  $i$ -ésimo grupo andará.

Para  $0 \leq i < Q$ , tu procedimiento deberá encontrar el número de posibles rutas con exactamente  $G[i]$  caminos que el  $i$ -ésimo grupo puede usar para llegar a la fuente  $P$ . Para cada grupo  $i$ , tu procedimiento deberá llamar `answer(X)` para reportar que el número de rutas en esa pregunta es  $X$ . Las respuestas deberán ser dadas en el mismo orden que los grupos. Si no hay rutas válidas tu procedimiento deberá contestar con  $X=0$ .

## Ejemplos

### Ejemplo 1

Considera el caso mostrado en la Figura 1, donde  $N=6$ ,  $M=6$ ,  $P=0$ ,  $Q=1$ ,  $G[0]=3$ , y

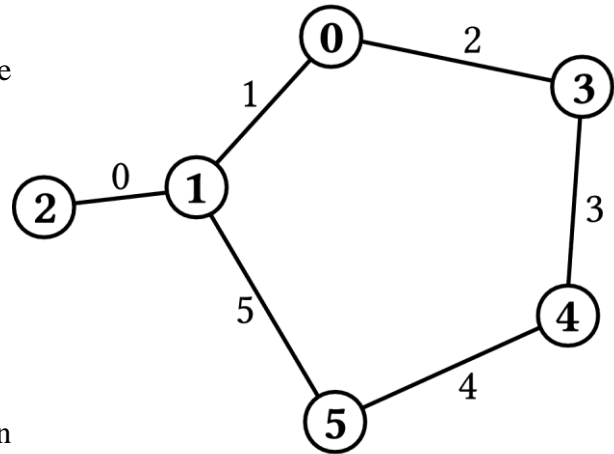
$$R = \begin{matrix} 1 & 2 \\ 0 & 1 \\ 0 & 3 \\ 3 & 4 \\ 4 & 5 \\ 1 & 5 \end{matrix}$$


Figura 1.

Nota que los caminos están listados en orden decreciente de belleza. Esto es, el camino 0 es el más bello de todos, el camino 1 es el segundo más bello y así.

Existen sólo dos rutas posibles de longitud 3:

- ▲  $1 \rightarrow 2 \rightarrow 1 \rightarrow 0$  y
- ▲  $5 \rightarrow 4 \rightarrow 3 \rightarrow 0$ .

La primera ruta inicia en la fuente 1. El camino más bello lleva de ahí a la fuente 2, al no haber alternativa, regresan por el mismo camino. De nuevo en la fuente 1, evitan el camino más bello por ser del que vienen y toman el camino que los lleva a la fuente 2. Terminando, al cabo de tres caminos en la fuente  $P=0$ .

El procedimiento deberá entonces llamar `answer(2)`.

### Ejemplo 2

Considera el caso que se muestra en la Figura 2, donde  $N=5$ ,  $M=5$ ,  $P=2$ ,  $Q=2$ ,  $K[0]=3$ ,  $K[1]=1$  y

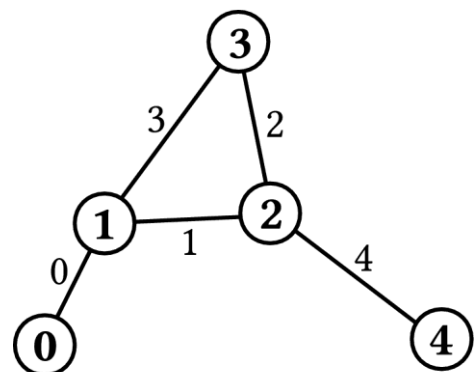
$$R = \begin{matrix} 1 & 0 \\ 1 & 2 \\ 3 & 2 \\ 1 & 3 \\ 4 & 2 \end{matrix}$$


Figura 2.

Para el primer grupo, sólo hay una forma válida de llegar a la fuente 2 en 3 pasos:  $1 \rightarrow 0 \rightarrow 1 \rightarrow 2$ .

Para el segundo grupo, hay dos formas válidas de llegar a 2 en 1 paso:  $3 \rightarrow 2$ , and  $4 \rightarrow 2$ .

Por lo tanto, una implementación correcta de `count_routes` debe llamar primero `answer(1)` para reportar la respuesta al grupo  $G[0]$  y después llamar `answer(2)` para reportar la respuesta al segundo grupo  $G[1]$ .

## Subtareas

### Subtarea 1 (49 puntos)

- ♣  $2 \leq N \leq 1\,000$
- ♣  $1 \leq M \leq 10\,000$
- ♣  $Q = 1$
- ♣  $G[0]$  es un entero entre 1 y 100, inclusivo.

### Subtarea 3 (31 puntos)

- ♣  $2 \leq N \leq 150\,000$
- ♣  $1 \leq M \leq 150\,000$
- ♣  $1 \leq Q \leq 2\,000$
- ♣ Cada elemento de  $G$  es un entero entre 1 y 1 000 000 000, inclusivo.

### Subtarea 2 (20 puntos)

- ♣  $2 \leq N \leq 150\,000$
- ♣  $1 \leq M \leq 150\,000$
- ♣  $Q = 1$
- ♣  $G[0]$  es un entero entre 1 y 1 000 000 000, inclusivo.

## Detalles de implementación

### Límites

- ♣ Tiempo límite de CPU: 5 segundos
- ♣ Memoria límite: 256 MB

**Nota:** No hay límite explícito para el tamaño de la memoria de stack. La memoria de stack cuenta para la memoria total usada por el programa.

### Interface (API)

- ♣ Carpeta de implementación: garden/
- ♣ Archivo del concursante: garden.c o garden.cpp o garden.pas
- ♣ Interface del concursante: garden.h o garden.pas
- ♣ Interface del evaluador (Grader): gardenlib.h o gardenlib.pas
- ♣ Evaluador de prueba (sample grader): grader.c o grader.cpp o grader.pas
- ♣ Entrada de evaluador de prueba: grader.in.1, grader.in.2, ...

**Nota:** El evaluador de prueba lee la entrada en el siguiente formato:

- ♣ Línea 1:  $N, M$  y  $P$ .
- ♣ Líneas 2 a  $M+1$ : descripción de los caminos; por ejemplo la línea  $i+2$  contiene  $R[i][0]$  y  $R[i][1]$  separados por un espacio, para  $0 \leq i < M$ .
- ♣ Línea  $M+2$ :  $Q$ .
- ♣ Línea  $M+3$ : arreglo  $G$  como una secuencia de enteros separados por espacio.
- ♣ Línea  $M+4$ : arreglo de soluciones esperadas como una secuencia de enteros separados por espacio.
- ♣ Salida esperada del evaluador de prueba: grader.expect.1, grader.expect.2, ...
- ♣ Para este problema, cada uno de estos archivos debe contener el texto preciso “Correct.”