

Rescatando... al príncipe!

Contribución de Natalia Pérez y Laura Rivero

Descripción del problema

Hace mucho tiempo les contamos de la existencia de un bosque con claros conectados por senderos (**Rescatando a la princesa - certamen Selección 2008**). Les hablamos de un príncipe que tenía que salvar a una princesa evitando ser interceptado por dragones (los dragones solo podían atacar al príncipe en un claro). También les informamos que los senderos eran bidireccionales y que todos los claros eran alcanzables desde cualquier punto.

Hoy podemos contarles felizmente que princesa ya fue rescatada pero los dragones planean un nuevo ataque: esta vez intentarán atrapar al príncipe.

Los dragones descansan en claros pero ni bien huelen la presencia de hombres en el bosque se ponen en movimiento —lo que sucede en el mismo instante en que el príncipe sale del castillo para comenzar su viaje—. Han decidido trasladarse constantemente a través de los senderos buscando interceptarlo en un claro (este movimiento lo realizarán siempre que puedan ya que, en algunos senderos, la vegetación ha crecido tanto que no les permite pasar). Han entendido que deben llegar al claro exactamente en el mismo momento que el príncipe para poder atraparlo, porque si no éste puede verlos y huir.

La princesa está preocupada ya que desde el castillo observa lo que ocurre en el bosque. Puede indicar exactamente cuáles son los claros en los que se encuentran cada uno de los dragones al momento de la salida del príncipe. Requiere de tu ayuda escribiendo la función:

príncipe(*C*, *S*, **Senderos**, *D*, **Dragones**, *CF*, **Cuales**)
que retorne los claros que debe ir alcanzando el príncipe para llegar sano y salvo a destino lo más rápido posible, sabiendo que: todos los senderos se recorren en la misma cantidad de tiempo —por lo cual tanto el príncipe como los dragones llegan a un claro en el mismo instante si parten de claros adyacentes al mismo—, el príncipe puede desplazarse por cualquier

sendero y que el castillo se ubica en el claro 1.

En caso de empate, cualquier combinación exitosa es válida. En caso de no existir un camino seguro, se te pide que indiques el conjunto de claros alcanzables por el príncipe con caminos seguros ordenados ascendentemente por número de claro.

Sus parámetros son:

C : cantidad de claros ($2 \leq C \leq 100.000$)

S : cantidad de senderos
($1 \leq S \leq 600.000$)

Senderos : arreglo de registros, de tamaño *S*, con los siguientes datos:

❖ **extremos**: arreglo con dos números que indican los extremos del sendero ($1 \leq \text{extremo} \leq C$).

❖ **ancho**: un número que indica el ancho del sendero luego de la crecida de la vegetación ($1 \leq \text{ancho} \leq 100$).

D : cantidad de dragones ($0 \leq D \leq 100$)

Dragones : arreglo de tamaño **D** con los siguientes datos:

❖ **inicio**: contiene el claro donde se ubica el dragón al momento de la salida del príncipe ($1 \leq \text{inicio} \leq C$).

❖ **dragonAncho**: un número que indica el ancho mínimo que debe tener un sendero para que el dragón pueda recorrerlo ($1 \leq \text{dragonAncho} \leq 100$).

CF : número de claro destino del príncipe ($1 \leq CF \leq C$)

Cuales : un arreglo para retornar la secuencia de números de claros que describe la trayectoria que deberá atravesar el príncipe para llegar lo más rápido posible a destino —incluyendo el castillo y la llegada—. En caso de no existir un camino seguro hasta el destino, se deberá indicar el conjunto de claros alcanzables por el príncipe con caminos seguros, ordenados numéricamente.

La función devuelve el largo llenado en **Cuales**

Nota: puede ocurrir que dos dragones crucen el mismo sendero en dirección contraria. Como saben saltar rápidamente solo basta que uno de ellos lo haga para no chocarse con el otro. También es

probable que se crucen con el príncipe, pero como los dragones necesitan más

espacio para poder atacar, el príncipe no corre peligro en esta circunstancia.

Ejemplo

Ver ejemplo en el anexo

Detalles de implementación

En un único archivo, llamado **principio.cpp**, **principio.c**, o **principio.pas** debes enviar una función que implante la función descrita antes, usando los siguientes prototipos:

En C/C++:

```
struct SENDERO { long extremos[]; long ancho; };
struct DRAGON { long inicio; long anchoDragon;};
long principio( long C, long S, SENDERO Senderos[], long D, DRAGON
Dragones[], long CF, long Cuales[] ) ;
```

En Pascal:

```
type SENDERO = array[1..600000] of record extremos:
array[0..1] of longint ; ancho : longint end ;
DRAGON = array[1..100] of record inicio : longint,
anchoDragon : longint end ;
CUAL = array[0..100000] of longint ;
function principio( C, S, D, CF : longint ; Senderos : var SENDERO
; var Dragones: DRAGON ; ; var Cuales CUAL ) ;
```

Evaluador local

El evaluador local (programa para probar ejemplos propios) lee la entrada por stdin en el siguiente formato:

- Línea 1: C(claros), S (senderos) y CF(destino)
- Línea 2+i ($0 \leq i < S$): 3 enteros separados por blanco indicando los extremos y ancho de cada sendero.
- Línea 2 + S: D(dragones)
- Línea 3+S+k ($0 \leq k < D$): 2 enteros que indican el número del claro donde se ubica cada dragón y su ancho

La función entrega el resultado por consola.:

En una línea el largo (valor de la función)

Luego el vector cuales plegado a razón de 10 claros por línea

Para el caso del ejemplo 1 la entrada podría ser:

```
# claros, senderos, destino
9 12 8
# extremos y anchos de senderos
1 2 9
1 5 2
1 3 8
2 9 6
```

```
2 3 1
4 3 2
5 4 1
4 7 9
7 8 3
9 6 2
6 8 7
8 3 8
# cantidad de dragones
2
# claros y anchos de dragones
4 2
6 8
```

Y la salida:

```
4
1 2 3 8
```

Para el caso del ejemplo 2 la entrada sería:

```
# claros, senderos, destino
9 12 8
1 2 9
1 5 2
1 3 8
```

```
2 9 6
2 3 10
4 3 2
5 4 11
4 7 9
7 8 3
9 6 20
6 8 7
8 3 8
# cantidad de dragones
3
4 2
```

```
6 8
7 3
```

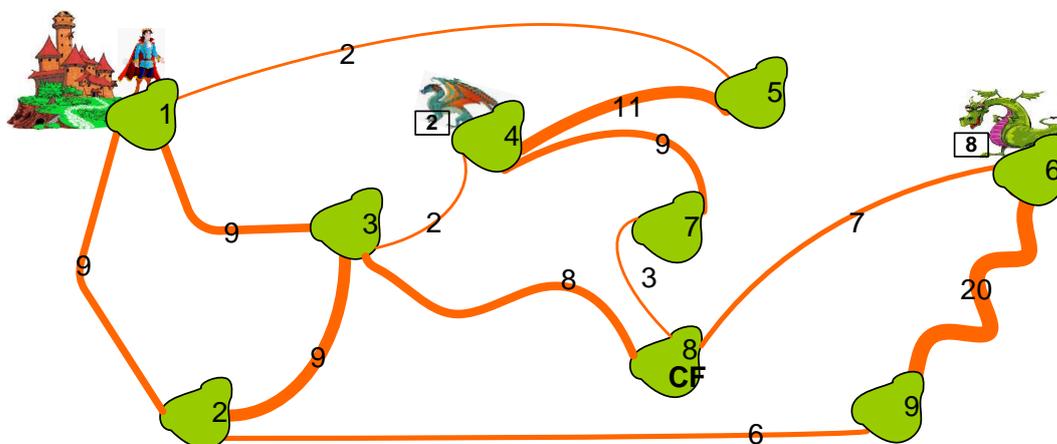
Y la salida:

```
4
1 2 9
```

Anexo:

Ejemplo 1: Hay un camino seguro

Si se recibieran los siguientes datos:



El príncipe puede llegar seguro a destino recorriendo los claros 1-2-3-8

Ejemplo 2: No hay un camino seguro

En cambio, con la siguiente ubicación de los dragones, el príncipe solo puede alcanzar los claros 1, 2 y 9 sin ser atacado por ningún dragón.

