

Equipos de comunicación

Contribución de Sergio Porter y Carlos Mendioroz

Descripción del problema

En un centro de cómputos moderno, al encargado de comunicaciones le han asignado la tarea de mejorar la confiabilidad de la infraestructura.

Sucintamente, el centro tiene un conjunto de armarios numerados (racks) con un conmutador cada uno (switch) que están interconectados a través de un conjunto de fibras ópticas.

La cantidad de fibras es más que suficiente para que dos conmutadores puedan comunicarse ya sea directamente, ya sea pasando el mensaje por otros que actúan de intermediarios. La lógica de las comunicaciones hace que los mensajes sigan el camino más corto. Esto es así porque la distancia recorrida afecta directamente la capacidad de comunicación. Todas las fibras están tabuladas con su largo, y se sabe que el camino más corto es único para cualquier origen y destino.

Lo que el EdC (encargado de comunicaciones) acaba de aprender es que los conmutadores calientan más cuanto más comunicaciones cursan, y ha decidido comprar un ventilador adicional para enfriar el armario donde se encuentra el conmutador más cargado como forma de evitar fallas.

El problema consiste en determinar cuál es el conmutador más cargado, suponiendo que las comunicaciones entre distintos armarios es uniforme, es decir, todos los armarios mantienen igual cantidad de comunicaciones con todos los demás. La carga del conmutador está dada por la cantidad de comunicaciones en las que está involucrado por ser terminal o intermediario de las mismas.

Se te pide que ayudes al EdC a decidir cuál es el conmutador más cargado, escribiendo la función:

```
temperatura( N, M, Origen,
             Destino, Largo)
que determine el número de armario en el
```

que se debería instalar el ventilador. De haber más de un conmutador igualmente cargado, cualquiera vale.

Sus parámetros son:

N : cantidad de armarios ($1 \leq N \leq 300$)

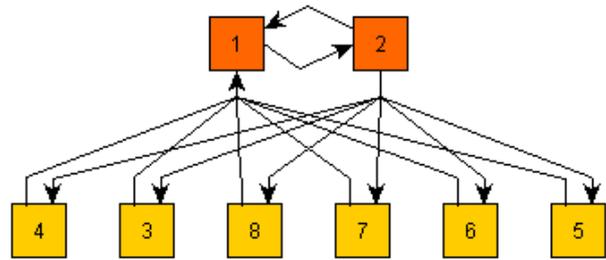
M : cantidad de fibras. ($1 \leq M \leq 60.000$)

Origen : arreglo de tamaño **M** con el número de armario de origen de cada fibra.

Destino : arreglo de tamaño **M** con el número de armario de destino de cada fibra.

Largo : arreglo de tamaño **M** con el largo de cada fibra.

Ejemplo



Una arquitectura bastante común es la que se denomina Leaf & Spine, una variante de Clos. Hay conmutadores de primer nivel totalmente interconectados que a su vez conectan con conmutadores de segundo nivel. Un ejemplo de 2x8 estaría dado por:

$N = 8$ $M = 14$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Origen	1	2	2	2	2	2	2	3	4	5	6	7	8	2
Destino	2	3	4	5	6	7	8	1	1	1	1	1	1	1
Largo	1	2	2	2	2	2	2	2	2	2	2	2	2	3

Detalles de implementación

En un único archivo, llamado `temperatura.cpp`, `temperatura.c`, o `temperatura.pas`. debes enviar una función que implemente la función descrito antes usando los siguientes prototipos:

```
En C/C++ int temperatura( int N, int M,
                          int Origen[], int Destino[], int Largo[])
```

Debes incluir también un archivo de encabezamiento **temperatura.h**

En Pascal function temperatura(N, M : longint ;

Origen, Destino, Largo: array[0..5999] of longint) : longint

Evaluador local

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

Línea 1: dos enteros N y M separados por un espacio

Línea $2+i$ ($0 \leq i < M$): 3 enteros separados por espacios que describen la i -ésima fibra dando el

Origen (un entero entre 1 y N), Destino (un entero entre 1 y N), Largo (un entero entre 1 y 50.000)

y entrega por consola el resultado de la función.

Para el caso del ejemplo la entrada sería:

```
8 14
1 2 1
2 3 2
2 4 2
2 5 2
2 6 2
2 7 2
2 8 2
3 1 2
4 1 2
5 1 2
6 1 2
7 1 2
8 1 2
2 1 3
```

Y la salida por consola podría ser:

```
1
```

Subtareas

Habrá casos de prueba por un total de 30 puntos en los cuales por toda fibra en un sentido habrá una en sentido contrario de igual longitud formando toda la red un árbol.

Habrá otro conjunto de casos de prueba por un total de 30 puntos donde N no superará 230