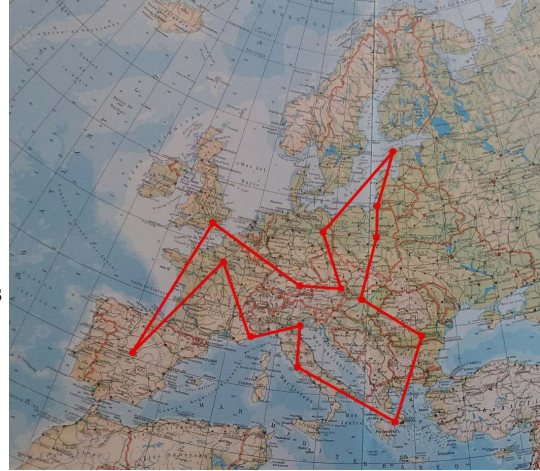


# Planificando las vacaciones

Estas vacaciones quiero hacer un circuito por Europa visitando las principales ciudades. He seleccionado unas cuantas que todavía no conozco y tengo intención de visitar todas ellas. Como no me gusta viajar de noche, me desplazaré de una ciudad a otra durante la tarde, dormiré en un buen hotel de la ciudad y aprovecharé las mañanas para hacer turismo. Cada noche dormiré en una ciudad distinta, así que sólo tengo un día para visitarla.

He preparado una ruta, que pasa por todas las ciudades sin repetir ninguna de ellas. Tengo el precio del billete más barato para ir de una ciudad a otra, así como el precio del hotel que más me gusta en cada ciudad. Pero no me llega con el dinero que he ahorrado este invierno y no quiero renunciar a ir a ninguna ciudad ni bajar la calidad de los hoteles que he elegido. He pedido consejo a mis amigos, y uno de ellos me ha indicado que si no me importa cambiar el orden en que visito las ciudades es posible que encuentre una ruta más barata, al utilizar trayectos de menor coste, o las ofertas de precios de los hoteles.



Ha prometido ayudarme si le paso el precio de los billetes entre cada ciudad, el precio de los hoteles en temporada alta y en temporada baja y cuantos días pasan desde que salgo hasta que comienza la temporada baja en los hoteles europeos. Le he recordado que debe incluir el precio de volver a casa desde la última ciudad, no vaya a ser que me quede sin fondos antes de tiempo.

Por ejemplo, en la figura del enunciado, si salgo de Madrid y voy a París, donde paso una noche, A continuación bajo a Niza, y luego Venecia y Roma... si la temporada baja comienza el día 3, esto significa que utilizaré el hotel de París (día 1) y Niza (día 2) con precio de temporada alta y Venecia (día 3) será la primera ciudad donde tenga el precio de temporada baja.

## *Requisitos de implementación.*

El problema se debe resolver utilizando la técnica de vuelta atrás. Se debe podar el árbol de ejecución impidiendo realizar llamadas recursivas que no darán soluciones en base a la solución ya construida, pero no es necesario realizar estimaciones sobre los días que todavía no han sido tratados.

Explicar el algoritmo: como es la solución que se va construyendo y como la vamos a ir construyendo. Poner comentarios en el código.

## Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de varias líneas. En la primera se muestra el número de ciudades  $n$  que voy a visitar (entre las que no se incluye mi ciudad de origen), el día del viaje en que comienza la temporada baja y el presupuesto que tengo. En la línea siguiente se dan los precios de los hoteles de las  $n$  ciudades en temporada alta y en la siguiente línea el precio de los hoteles en la temporada baja. Por último en las  $n+1$  líneas siguientes se da el precio de los billetes entre cada par de ciudades, incluyendo la mi ciudad de origen, que se considera la ciudad 0. El precio de ir de la ciudad A a la B será el mismo que el de ir de la ciudad B a la A. La entrada de datos termina con un caso con tres ceros.

El número de ciudades es un entero,  $2 \leq n \leq 20$  y el coste de los billetes y de los hoteles  $10 \leq p \leq 500$ .

## Salida

Para cada caso de prueba se muestra en una línea el precio de la ruta más barata. Si no existe ninguna ruta posible con el presupuesto que tengo se escribirá *No puedes ir*.

### Entrada de ejemplo

```
1 1 100
150
80
0 10
10 0
4 3 200
1 1 1 1
1 1 1 1
0 30 20 10 50
30 0 15 40 60
20 15 0 16 40
10 40 16 0 35
50 60 40 35 0
4 3 350
50 100 150 70
40 50 70 60
0 30 20 10 50
30 0 15 40 60
20 15 0 16 40
10 40 16 0 35
50 60 40 35 0
0 0 0
```

### Salida de ejemplo

```
100
134
No puedes ir
```

### Una combinación de ciudades visitadas con el precio dado

```
100
0 1

134
0 1 2 4 3

No puedes ir
0 4 1 2 3
```