

Normas de realización del examen

1. Debes desarrollar e implementar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc>.
2. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen.
3. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
4. Del enlace **Material para descargar** puedes descargar un archivo comprimido que contiene material que puedes utilizar para la realización del examen (transparencias de clase, implementación de las estructuras de datos, una plantilla de código fuente y ficheros de texto con los casos de prueba de cada ejercicio del enunciado).
5. Los ficheros con las implementaciones de las estructuras de datos están instalados en el juez, por lo que no es necesario subirlos como parte de tu solución (y conviene no hacerlo).
6. Los ejercicios están identificados con el nombre del tema de la asignatura en el que habrían aparecido si hubieran sido propuestos como parte de los ejercicios de la evaluación continua. Para obtener la máxima puntuación, las soluciones deberán seguir los criterios exigidos a los ejercicios de ese tema durante el curso (en cuanto a encapsulación, eficiencia, simplicidad, análisis de costes, etc.).
7. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.
8. Las notas de los ejercicios suman 7 puntos. La calificación obtenida en este examen será sumada a la obtenida por la evaluación continua (sobre 3 puntos) para obtener la calificación final de la asignatura.
9. Al terminar el examen, dirígete al puesto del profesor y rellena con tus datos la hoja de firmas que él tendrá. Muéstrale tu documento de identificación.

Primero resuelve el problema. Entonces, escribe el código.

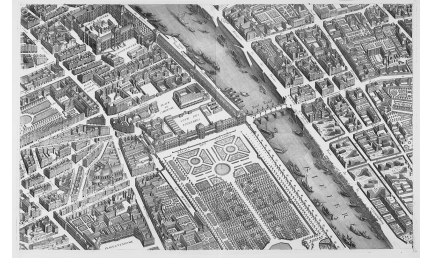
— John Johnson

*Comentar el código es como limpiar el cuarto de baño;
nadie quiere hacerlo, pero el resultado es siempre
una experiencia más agradable para uno mismo y sus invitados.*

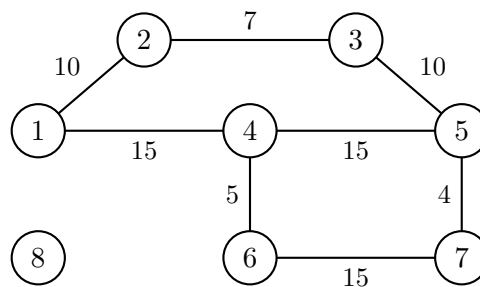
— Ryan Campbell

Ejercicio 1. Grafos y estructuras de partición (4 puntos)

Hace poco me he mudado a una nueva ciudad. Aprovechando que me gusta caminar, siempre que puedo voy andando a los sitios y así voy conociendo la ciudad. Cuando voy lejos y tengo prisa suelo coger las grandes avenidas porque las conozco más y sé que no voy a perderme, pero soy consciente de que, muchas veces, callejeando por calles cortas recorrería menos distancia, aunque tuviese que atravesar muchas de ellas. Me pregunto cuántas veces el camino más corto en distancia también es el que pasa por menos calles.



Por ejemplo, si el siguiente esquema representa la ciudad, con 8 intersecciones y 8 calles, donde junto a cada calle aparece su longitud medida en metros, para ir del punto 1 al punto 5 el camino más corto es el $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$, que recorre 27 metros y atraviesa tres calles, mientras que el camino $1 \rightarrow 4 \rightarrow 5$, aunque es más largo (30 metros) pasa solamente por dos calles. En cambio, el camino más corto que une los puntos 6 y 5 sí utiliza el menor número de calles (no hay ningún otro camino con menos calles).



Entrada

La entrada consta de varios casos de prueba, ocupando cada uno de ellos varias líneas.

En la primera aparece el número N (entre 1 y 10.000) de intersecciones en la ciudad, y en la segunda el número C (entre 0 y 100.000) de calles (entre intersecciones). A continuación, aparece una línea por cada calle con tres enteros, que indican los números de las intersecciones que une la calle (números entre 1 y N) y su longitud (un valor entre 1 y 5.000) medida en metros. Todas las calles pueden recorrerse en ambos sentidos.

A continuación aparece el número K de consultas (no más de 10) seguido de esas consultas: dos números que representan las intersecciones origen y destino. Se garantiza que para cada consulta, el camino más corto entre origen y destino es único.

Salida

Para cada caso de prueba se escribirá una línea por cada consulta que contendrá la distancia, medida en metros, del camino más corto que conecta el origen con el destino, seguida de la palabra **SI** si ese camino además atraviesa el menor número de calles o **NO** en caso contrario. Si para una consulta no existiera camino que conecte el origen con el destino, entonces se escribiría **SIN CAMINO**.

Después de la salida de cada caso se escribirá una línea con ----.

Entrada de ejemplo

```
8
8
1 2 10
2 3 7
3 5 10
1 4 15
4 5 15
4 6 5
5 7 4
6 7 15
3
1 5
3 8
6 5
```

Salida de ejemplo

```
27 NO
SIN CAMINO
19 SI
----
```

Ejercicio 2. Programación dinámica (3 puntos)

Para desayunar, mi hermana gemela y yo solemos tomar un bizcocho adornado con frutas escarchadas por encima. Nuestras frutas preferidas son la naranja y el limón. Los mejores días son aquellos en que las dos tomamos cada una un trozo que tenga naranja por encima o bien las dos tomamos trozos con limón, es decir, las dos probamos el mismo sabor. Cada día cortamos dos trozos iguales de bizcocho, pero nuestra madre solamente nos deja cortarlo de alguno de los dos extremos del bizcocho, los dos del mismo lado o cada uno de un lado distinto, pero no podemos partirlo y sacar un trozo del centro. Teniendo en cuenta que nunca hay dos trozos de bizcocho consecutivos que tengan la misma fruta, ¿cuál es el máximo número de días en que podremos tomar dos trozos con la misma fruta por encima?



Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número N , siempre par, de trozos de bizcocho que se pueden cortar en total ($2 \leq N \leq 1.000$). En la línea siguiente aparecen N números, que indican la fruta que tiene encima cada trozo del bizcocho, empezando por el de la izquierda. El valor 0 indica que no tiene fruta, el valor 1 indica que tiene naranja y el valor 2 indica que tiene limón.

Salida

Para cada caso de prueba se escribirá en una línea el número máximo de veces que podemos tomar dos trozos con la misma fruta.

Entrada de ejemplo

```
4
1 2 1 2
8
0 1 2 0 0 2 1 2
12
1 2 0 0 2 0 2 1 0 0 2 0
```

Salida de ejemplo

```
0
2
2
```