

Técnicas algorítmicas en ingeniería del software

Grado en Ingeniería del Software (UCM)

EXAMEN FINAL DE SEPTIEMBRE

Curso 2015/2016

Ejercicio 1. Grafos y estructuras de partición (3 puntos)

Recorriendo el archipiélago en bicicleta

Al archipiélago *Milislotes* han llegado las bicicletas. Sus habitantes están estusiasmados y no se bajan de la bici en todo el día, yendo de un lado para otro sin descanso. Pero las islas son tan pequeñas que terminan todos mareados de dar vueltas sin parar, por lo que se están planteando crear una red de puentes rectos que les permita ir en bici de cualquier isla a cualquier otra.



Han pedido presupuesto al arquitecto, que ha confeccionado una lista con todos los puentes que podrían construirse entre islas del archipiélago y cuánto costaría construir cada uno de esos puentes. ¿Podrías ayudarles a decidir qué puentes construir de tal forma que se pueda ir en bici desde cualquier isla a cualquier otra y el coste total de la obra sea lo mínimo posible?

Entrada

La entrada está formada por varios casos de prueba. Cada uno ocupa varias líneas: en la primera aparece el número I de islas en el archipiélago (entre 1 y 1.000); en la siguiente aparece el número P de puentes presupuestados (entre 0 y 10.000); y a continuación aparece una línea por cada uno de estos puentes con tres enteros, las islas que une (numeradas entre 1 e I) y el coste de construir ese puente (un valor entre 1 y 100.000). Los puentes siempre van de una isla a otra distinta, son transitables en bici en ambos sentidos, y no se ha presupuestado más de un puente entre un mismo par de islas.

Salida

Para cada caso de prueba se escribirá, en una línea, el coste mínimo de construir los puentes necesarios para unir todas las islas por bici. Si no hubiera suficientes puentes presupuestados para lograrlo, se escribirá `No hay puentes suficientes`.

Entrada de ejemplo

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

Salida de ejemplo

```
14
No hay puentes suficientes
```

Ejercicio 2. Algoritmos voraces (2 puntos)

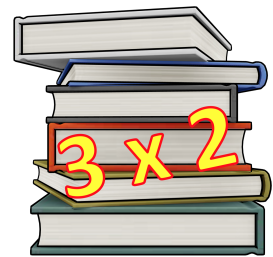
Oferta 3×2

En mi librería favorita, un día al año sacan una oferta de 3×2 : por cada tres libros que compras te regalan uno, el de menor precio. Yo espero ansioso a que llegue ese día y entonces compro todos los libros que puedo.

El año pasado compré libros que valían 40, 35, 30, 25, 20, 15 y 10 euros, por los que pagué 150 euros, obteniendo un descuento total de 25 euros, ya que la librería utiliza un sistema estricto (¡y muy ventajoso para ella!) a la hora de seleccionar los libros que te regala: son siempre los de menor precio del lote.

Pero después me di cuenta de que si hubiera ido a comprar varias veces podría haber obtenido un descuento mayor. Por ejemplo, si primero hubiera comprado los libros con precios 40, 35, 30 y 25, me habrían regalado el de 25, y al comprar después el resto me hubieran regalado el que valía 10, obteniendo un descuento total de 35 euros.

¿Puedes ayudarme a decidir cómo debería comprar los libros que quiero este año para ahorrarme lo máximo posible?



Entrada

El programa deberá encontrar la solución a diferentes casos de prueba. Cada caso consta de dos líneas. En la primera aparece el número (entre 1 y 1.000) de libros que quiero comprar. En la segunda aparecen los precios de los libros (entre 1 y 10.000), separados por espacios.

Salida

Para cada lote de libros se escribirá el descuento máximo que puedo obtener si me aprovecho de repartirlos en varias compras.

Entrada de ejemplo

```
7
40 35 30 25 20 15 10
3
50 10 30
2
25 20
```

Salida de ejemplo

```
45
10
0
```

Ejercicio 3. Programación dinámica (3 puntos)

Construyendo dianas

Del juego de los dardos hay muchas variantes. Todas consisten en tirar los dardos a una diana dividida en sectores de diferentes colores o que tienen asociada una puntuación distinta y conseguir cierto objetivo.



Queremos proponer una variante a la liga de dardos local: conseguir sumar cierto valor a partir de las puntuaciones obtenidas con los dardos pero con el menor número de tiros que sea posible. Por ahora tenemos una serie de dianas a las que hemos asignado puntuaciones diferentes a sus sectores y estamos interesados en conocer si ciertos valores pueden ser conseguidos tirando dardos a esas dianas y cuántos dardos como mínimo son necesarios en cada ocasión.

Entrada

En la entrada aparecerán diferentes configuraciones de dianas y objetivos. Cada una ocupa dos líneas. En la primera aparecen dos números: el valor (entre 1 y 500) que hay que conseguir sumar tirando dardos a una diana y el número S de sectores (entre 1 y 50) en los que está dividida la diana. En la segunda línea aparecen, en orden creciente, las S puntuaciones asociadas a esos sectores (valores entre 1 y 500).

Salida

Para cada caso se escribirá el menor número de dardos necesarios para conseguir la cantidad, separado por dos puntos de las puntuaciones que permiten conseguir ese valor, ordenadas de mayor a menor y separadas por espacios.

Si hay varias soluciones, se escribirá aquella cuya mayor puntuación sea la más alta; si aún siguen existiendo varias soluciones, se escribirá aquella cuya segunda mayor puntuación sea la más alta; y así sucesivamente.

Si es imposible conseguir el objetivo con las puntuaciones asignadas a los sectores de la diana, se escribirá *Imposible*.

Entrada de ejemplo

```
100 5
10 15 20 25 30
8 3
1 4 6
25 2
8 12
```

Salida de ejemplo

```
4: 30 30 30 10
2: 4 4
Imposible
```

Normas de realización del examen

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc.fdi.ucm.es/domjudge/team>.
2. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
3. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
4. Puedes descargar el fichero <http://exacrc.fdi.ucm.es/domjudge/team/material-sep.zip> que contiene material que puedes utilizar para la realización del examen (diapositivas 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 8 puntos. La calificación obtenida en este examen será sumada a la obtenida por la evaluación continua (sobre 2 puntos) para obtener la calificación final de la asignatura.