

Técnicas algorítmicas en ingeniería del software

Grado en Ingeniería del Software (UCM)

EXAMEN FINAL DE JULIO

Curso 2018/2019

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.

Ejercicio 1. Colas de prioridad y montículos (2.5 puntos)

Desde hace algún tiempo, quedamos la pandilla para ir a volar drones los sábados. Cada dron necesita dos pilas para poder volar: una de 9 V y otra de 1.5 V. En el club tenemos dos cajas para guardar las pilas, en una tenemos las de 9 V y en otra las de 1.5 V. Cada sábado cogemos las que están más cargadas y las colocamos en los drones. Para aprovechar al máximo el tiempo de vuelo, colocamos siempre las dos pilas más cargadas de cada tipo juntas, ya que los drones solo vuelan mientras las dos pilas tienen carga; después las dos siguientes más cargadas las ponemos en el siguiente dron; y así mientras queden pilas con carga de los dos tipos. Una vez colocadas las pilas, echamos los drones a volar. Cuando todos ellos acaban en el suelo por agotamiento de alguna de sus pilas, volvemos al club y guardamos en las cajas las pilas que todavía no están totalmente gastadas.

Por ejemplo, si a un dron le pusimos una pila de 9 V que permitía volar 5 horas y una pila de 1.5 V que permitía volar 2, el dron habrá volado 2 horas y al volver al club guardaremos la pila de 9 V a la que le quedarán 3 horas de vuelo. La pila de 1.5 V estará agotada y la echaremos al cubo de reciclaje.

Queremos saber cuántas horas de vuelo realizarán entre todos los drones cada sábado que podamos salir a volar, antes de que se agoten las pilas que hay ahora mismo en las cajas. Las pilas las tenemos que colocar en el club, por lo que cada dron solo puede volar una vez cada sábado.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número N de drones que tenemos ($1 \leq N \leq 1.000$), el número A de pilas de 9 V y el número B de pilas de 1.5 V ($1 \leq A, B \leq 200.000$). En la línea siguiente se indica el número de horas de vuelo que permite la carga de cada una de las pilas de 9 V y en la tercera línea el número de horas de vuelo que permite cada una de las pilas de 1.5 V.

Salida

Para cada caso de prueba se escribirá en una línea el número de horas de vuelo que realizarán los drones cada sábado, mientras se pueda salir a volar algún dron.

Se garantiza que esos números nunca serán mayores que 10^9 .

Entrada de ejemplo

```
2 4 2
5 12 7 15
10 10
2 4 3
5 12 7 15
20 20 2
3 3 3
25 15 10
20 20 5
1 4 6
5 9 2 6
7 3 3 1 6 4
```

Salida de ejemplo

```
20
27 12
40 5
7 6 4 2 2 1
```

Ejercicio 2. Grafos y estructuras de partición (2 puntos)

En una red social hay una serie de usuarios que se comunican entre ellos dentro de una serie de grupos de amigos. Queremos analizar el proceso de distribución de noticias entre estos usuarios.

Inicialmente, algún usuario recibe una noticia de una fuente externa. Entonces envía la noticia a sus amigos en la red (dos usuarios son amigos si hay al menos un grupo al que pertenezcan ambos). Los amigos envían a su vez esa noticia a sus amigos y el proceso continua así hasta que no exista una pareja de amigos tal que uno de ellos conozca la noticia y el otro no.

Para cada usuario de la red, queremos saber cuántos usuarios terminarían conociendo la noticia si inicialmente solamente ese usuario la conocía.

Entrada

La entrada está formada por una serie de casos, cada uno de los cuales ocupa varias líneas. En la primera línea de cada caso aparecen dos números: el número N de usuarios de la red y el número M de grupos ($1 \leq N, M \leq 100.000$). A continuación aparecen M líneas describiendo esos grupos. Para cada grupo, la descripción comienza con el número de usuarios del grupo (entre 0 y N) seguido de los identificadores de esos usuarios (todos distintos), números entre 1 y N . La suma de los tamaños de todos los grupos no es mayor que 500.000.

Salida

Para cada caso se escribirá una línea con N números. El número i -ésimo indicará el número de usuarios que terminarían conociendo la noticia si el usuario i fuera quién comenzara a distribuirla.

Entrada de ejemplo

```
7 5
3 2 5 4
0
2 1 2
1 1
2 6 7
4 2
1 1
1 3
```

Salida de ejemplo

```
4 4 1 4 4 2 2
1 1 1 1
```

Ejercicio 3. Programación dinámica (2.5 puntos)

La empresa *Illuminate, S.L.* está especializada en la iluminación de salas de fiesta. Coloca tiras de bombillas en el techo y luces indirectas en las paredes, hasta obtener una iluminación que satisfaga a los organizadores de las fiestas. Para ser más competitiva se ha asociado con una fábrica que le proporciona tantas bombillas como necesite, a bajo coste, pero solo de algunas potencias.

Para que los organizadores queden satisfechos, les pide en el contrato que indiquen la potencia instalada que desean para la sala. Sin embargo, se han dado cuenta de que muchas veces sería más barato realizar una instalación que superase la potencia requerida y piensan que esto no debería molestar a los organizadores, ya que la sala estará más iluminada. Por ello han cambiado el contrato y ahora piden al organizador que les indique una potencia mínima para la sala, pudiendo la empresa instalar una potencia superior siempre que no sobrepase la potencia máxima admitida por la instalación.

¿Sabrías calcular la potencia que debes instalar en la sala, sabiendo la potencia máxima admitida por la instalación y la potencia mínima requerida por los organizadores para que la instalación sea lo más barata posible?

Entrada

La entrada consta de una serie de casos de prueba. Cada caso consta de tres líneas. En la primera línea se indica el número N de tipos de bombillas ($1 \leq N \leq 1.000$), la potencia máxima PM_{\max} permitida y la potencia mínima PM_{\min} requerida por los organizadores ($1 \leq PM_{\min} \leq PM_{\max} \leq 1.000$). En la línea siguiente se da la potencia de cada tipo de bombilla que nos proporciona la fábrica y en la tercera línea se indica el coste de cada tipo de bombilla.

Salida

Para cada caso de prueba se escribirá en una línea el coste mínimo de la instalación y la potencia que se debe instalar. Si el coste mínimo se consigue con varias potencias, se escribirá la menor de todas ellas. Si es imposible conseguir ninguna de las potencias permitidas, se escribirá **IMPOSIBLE**.

Entrada de ejemplo

```
2 7 4
5 2
30 10
3 8 6
2 3 5
30 45 50
3 11 11
4 6 4
1 2 3
```

Salida de ejemplo

```
20 4
80 7
IMPOSIBLE
```