Técnicas algorítmicas en ingeniería del software

Grado en Ingeniería del Software (UCM)

12 de enero de 2023

Normas de realización del examen

1. Esta parte del examen dura **2 horas**.
2. Debes desarrollar e implementar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección `http://exacrc`. Utiliza para cada ejercicio el fichero plantilla que te hemos entregado, escribiendo tu solución en los espacios reservados para ello, siempre entre las etiquetas `<answer>` y `</answer>`.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. En la segunda parte del examen (el cuestionario) una pregunta te pedirá que digas el usuario que has usado. Es **muy importante** que la contestes bien.
4. Escribe tu **nombre y apellidos** en el hueco reservado para ello en el fichero plantilla de cada ejercicio.
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). Simplemente utiliza el `#include` correspondiente.
6. Puedes acceder a las diapositivas del curso en la dirección `http://exacrc/diapositivas`. Si necesitas consultar la documentación de C++, está disponible en `http://exacrc/cppreference`.
7. 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, partes a incluir en la explicación de la solución, análisis de costes, etc.).
8. 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.

Primero resuelve el problema. Entonces, escribe el código.

— John Johnson

Ejercicio 1. Grafos y estructuras de partición (3.5 puntos)

En una región con pueblos dispersos, tras una gran nevada, todas sus carreteras han quedado bloqueadas. Cada una de estas carreteras conecta dos pueblos (en ambos sentidos), y se ha estimado el tiempo en horas que se va a tardar en limpiar y volver a abrir cada una de las carreteras. Estos tiempos son independientes entre sí, ya que se trabaja en todas las carreteras a la vez.

¿Cuál es el mínimo número de horas que tienen que pasar para que todos los pueblos vuelvan a estar conectados? Es decir, para que para cualquier pareja de pueblos exista una ruta entre los dos pueblos que pase por una o más carreteras que ya han sido abiertas.

Entrada

La entrada comienza con el número de casos de prueba que aparecerán a continuación.

La primera línea de cada caso contiene dos números enteros: el número de pueblos N (entre 1 y 10.000) y el número de carreteras M (entre 0 y 100.000). A continuación aparecen M líneas, cada una describiendo una carretera: primero aparecen los números de los pueblos que la carretera conecta (los pueblos están numerados de 1 a N) y a continuación el número de horas estimadas que requerirá la limpieza de esa carretera (números entre 1 y 10.000).

Salida

Para cada caso de prueba se escribirá una línea con el número de horas que tienen que pasar para que los pueblos vuelvan a estar conectados. Si no es posible conectar todos los pueblos, se mostrará una línea con la palabra `Imposible`.

Entrada de ejemplo

```
3
3 3
1 2 10
2 3 20
3 1 30
4 2
1 3 10
4 2 10
4 3
1 2 20
1 3 30
1 4 10
```

Salida de ejemplo

```
20
Imposible
30
```

Ejercicio 2. Programación dinámica (3.5 puntos)

Marta tiene un dado y se divierte tirándolo varias veces seguidas y calculando la suma de los valores que le van saliendo (1 + 5 + 2 + 3 + 2 + …). Esta tarde llegó hasta 100, y al ser un número tan exacto, se preguntó de cuántas formas distintas podría haber llegado a esa misma suma. Cogió lápiz y papel y se puso a escribir las formas de conseguir 10 (quería empezar por algo más sencillo). Además, hace poco ha aprendido que la suma es conmutativa, por lo que no le interesa contar dos formas que solamente se diferencian en el orden en que han salido los números. Para ella, que le salga primero un 4 y luego un 6, es lo mismo que primero le salga el 6 y después el 4. Lleva un buen rato y ha contado 35 formas de conseguir la suma 10. ¿Serán todas?

Iba a ponerse con la difícil tarea de calcular las formas de conseguir 100, pero antes ha empezado a pensar qué ocurriría si el dado en vez de 6 caras tuviera k, numeradas del 1 al k. Eso parece aún más difícil. Necesita ayuda.

Entrada

La entrada comienza con el número de casos de prueba que vendrán a continuación. Cada caso ocupa una línea con dos números: el número k de caras del dado y la suma s que se quiere conseguir.

Salida

Para cada caso de prueba se escribirá una línea con el número de formas distintas (sin importar el orden en el que salgan los dados) de conseguir la cantidad s con un dado que tiene k caras, numeradas del 1 al k. El número de formas nunca será mayor que 10^9.

Entrada de ejemplo

```
3
6 10
6 100
10 100
```

Salida de ejemplo

```
35
189509
6292069
```