

CI-5651: Proyecto #1

Fecha de entrega: Viernes 23 de Enero de 2015

G. Palma & A. Flores

El Proyecto #1 de *CI-5651* está inspirado en los conjuntos de problemas de un *maratón de programación*. En este sentido, el proyecto consta de 3 problemas independientes, inspirados en problemas de la ACM-ICPC¹. Cada problema cuenta con un enunciado que describe en detalle el problema que se quiere resolver, así como el formato que sigue la entrada y la salida del programa que lo solucione. La *entrada* describe la instancia del problema que se quiere resolver, y la *salida* describe los resultados obtenidos para dicha instancia. Al igual que en los maratones de programación, para la evaluación de las soluciones al Proyecto #1 se usará la entrada y salida *estándar*.

¹ <http://icpc.baylor.edu/>

Problema 1

Es el año 1990, y la Dirección de Servicios Telemáticos (DST) de la Universidad Simón Bolívar (USB) desea instalar un nuevo sistema de conexión entre computadoras: *internet*. Para conectarse a internet, la universidad tiene asignado suficiente presupuesto para comprar M modems. Para que una oficina en la universidad pueda acceder a la red, debe tener un modem asignado, o estar conectada (directa o indirectamente) a una oficina con modem. Estas conexiones entre oficinas pueden hacerse usando cables regulares (cuando la distancia no exceda R metros, con un costo de U bolívars por metro) o cables especiales (cuando la distancia exceda R metros, con un costo de V bolívars por metro).

La DST te ha contratado como preparador para que los ayudes a solucionar este problema. Tu objetivo es minimizar los costos de conectar a internet a las N oficinas de la universidad.

Entrada La primera línea contiene el número de casos de prueba. La primera línea de cada caso de prueba contiene los valores N , R , M , U y V . Las siguientes N líneas contienen dos enteros x_i y y_i , que son las coordenadas de la i -ésima oficina. Donde...

$$N, M, R, U, V \in \mathbb{Z}^+$$

$$x_i, y_i \in \mathbb{Z}$$

$$M \leq N$$

$$U \leq V$$

Salida Por cada caso de prueba se debe imprimir una línea con el texto “Caso #i: A B”. Donde i es el número del caso de prueba (empezando en 1), y A y B (redondeados usando 3 decimales) son la suma de los costos totales de cable regular y especial respectivamente, necesarios para realizar una conexión de costo mínimo.

Entrada	Salida
2	Caso #1: 2.000 0.000
3 1 1 1 1	Caso #2: 4.000 6.000
0 0	
0 1	
1 0	
6 1 3 2 3	
0 0	
0 2	
2 0	
3 2	
2 3	
3 3	

Problema 2

Tras más de 20 años de servicio, la red interna de la USB requiere mucho mantenimiento. A medida que los diferentes modems se fueron dañando, la falta de presupuesto obligó a la universidad a modificar la infraestructura de la red interna. Actualmente, todas las oficinas de la universidad están directa o indirectamente conectadas con el único modem restante, que se encuentra en la DST. Más aún, no existen conexiones redundantes, por lo que existe **un** solo camino entre cada oficina. Como si no fuera suficiente, algunas de las conexiones entre oficinas han empezado a fallar. Como en la red no existen redundancias, la falla de uno de los cables provoca la desconexión de muchas oficinas.

La DST necesita tu ayuda para determinar el número de pares de oficinas que están desconectadas entre sí, para poder solicitar el presupuesto correspondiente. Es decir, contar el número de pares de oficinas $\{x, y\}$ tal que no existe un camino —a través de las conexiones de la red— entre las oficinas x y y . Más aún, quieren calcular este valor cada vez que una conexión se daña, para así mostrar al ministerio la gravedad de la situación.

Entrada La primera línea contiene el número de casos de prueba. Antes de cada caso de prueba, se ha dejado una línea en blanco. La primera línea de un caso de prueba contiene el número de oficinas $N \in \mathbb{Z}^+$. Las siguientes $N - 1$ líneas tienen dos enteros x_i y y_i , indicando que existe una conexión entre esas oficinas (1-indexadas). La siguiente línea contiene el número de queries C a realizar. Las siguientes C líneas describen cada query:

- **R** x indica que la conexión x (1-indexada respecto al orden de la entrada) se dañó.
- **Q** indica que debe calcularse (e imprimirse) el número de pares de oficinas desconectadas entre sí, en función del estado actual de la red.

Salida Una línea por cada query de tipo “Q”. Entre cada caso de prueba, debe imprimir una línea en blanco.

Entrada	Salida
2	0
	2
3	3
1 2	
1 3	0
5	3
Q	5
R 1	6
Q	
R 2	
Q	
4	
1 2	
1 3	
1 4	
7	
Q	
R 1	
Q	
R 2	
Q	
R 3	
Q	

Problema 3

En la USB, la Dirección de Admisión y Control de Estudios (DACE) ha tenido algunos problemas de organización. El bajo número de empleados y la enorme cantidad de actividades a realizar, han implicado un aumento considerable en las colas para el retiro de documentos.

Por esa razón, te han contratado para que los ayudes a mejorar la organización de sus actividades internas. Tu objetivo es calcular el máximo número de actividades que puede realizar cada empleado, del conjunto total de actividades que tiene asignadas. Cabe destacar, que una persona no puede realizar dos actividades al mismo tiempo.

Entrada La primera línea contiene el número de empleados en DACE. Luego se describen los intervalos de las actividades que tiene asignado cada empleado. La primera línea de dicha descripción contiene el número de actividades N asignadas a cada empleado. Las siguientes N líneas contienen dos enteros x_i y y_i ($x_i < y_i$), que representan el tiempo de inicio y fin de cada actividad asignada al empleado correspondiente.

Salida Por cada empleado, imprime el máximo número de actividades que pueden realizar.

Entrada	Salida
3	1
3	2
3 9	3
2 8	
6 9	
4	
1 7	
5 8	
7 8	
1 8	
6	
7 9	
0 10	
4 5	
8 9	
4 10	
5 7	

Instrucciones de Entrega

- Equipos de **máximo** 2 personas.
- **Un** archivo ZIP con los archivos correspondientes. El nombre del archivo comprimido debe contener los carnets de los integrantes separados por un guión. Ejemplo: 0910298-0910147.zip
 - La solución a cada problema debe estar en **un solo archivo** con el código fuente. Su nombre debe ser el número del problema. Por ejemplo: 1.cpp
 - Un archivo PDF llamado P1.pdf donde describan sus soluciones a cada problema. Una descripción del diseño del algoritmo que implementaron para la solución del problema, el **pseudocódigo**, y un análisis de su complejidad en tiempo y espacio.
- Las soluciones deben ser enviarse a `alejandroflores.af@gmail.com`.

Instrucciones de Implementación

- Los lenguajes de programación: C/C++ y Java.
- Su código debe compilar (y correr) en `www.ideone.com`.
 - **Un** solo archivo por problema.
 - Usen **solo** librerías estándar.
 - Usen las entradas y salidas estándar (*standard input* y *standard output*). Para probar su programa desde la terminal, pueden redireccionar la entrada y salida estándar a archivos. Ejemplo:

```
./1 < entrada1.txt > salida1.txt  
java 1 < entrada1.txt > salida1.txt
```
- El código va a ser revisado y evaluado (no solo la correctitud).
 - Debe estar debidamente documentado.
 - Debe cumplir con buenas prácticas de implementación (guías de estilo de C/C++ y Java).
 - La limitación a un solo archivo por problema no es excusa para la falta de modularidad y orden en el código.
- El formato de entrada y salida debe seguirse de forma estricta. Un espacio de más en la salida puede significar una evaluación incorrecta.