

0 OBJETIVOS

- Diseñar soluciones computacionales para problemas.
- Estimar costos de las soluciones planteadas.
- Implementar soluciones en *Java*.

Se premiarán las mejores soluciones y se castigarán las peores, en cuanto a eficiencia en tiempo y espacio.

1 CONDICIONES GENERALES

Hay tres problemas para resolver mediante soluciones implementadas en *Java*.

Para cada problema se pide:

- Análisis temporal y espacial.
- Una solución java.

2 PROBLEMAS

A Longitud de A-segmento más largo

Para un arreglo de enteros b , un *A-segmento* es un subarreglo para el que todos sus elementos tienen valor menor o igual al valor absoluto del elemento del extremo derecho del subarreglo.

Problema

Dado $b: [0..n-1] : \text{int}$, $n > 0$, se quiere encontrar la longitud del A-segmento más largo contenido en b .

Ejemplo:

Supóngase que $n=5$ y $b[0..4] = [3, 1, -1, 2, 0]$.

Son A-segmentos, entre otros:

- $b[2..2] = [-1]$, con longitud 1;
- $b[2..3] = [-1, 2]$, con longitud 2.
- $b[1..3] = [1, -1, 2]$, con longitud 3;

El tercer ejemplo corresponde, precisamente, a un A-segmento de máxima longitud dentro de b .

B Vigilantes

Una empresa de vigilancia tiene un contrato para cuidar una instalación industrial en la que se definen *puntos de control* para los vigilantes. El conjunto de puntos de control es $V = 1 \dots n$, con $n > 0$. Entre algunos pares de puntos de control -diferentes- se definen *senderos*, de modo que el par $(a, b) \in V \times V$ representa un sendero entre los puntos de control a y b . Todo sendero puede ser recorrido en cualquier sentido.

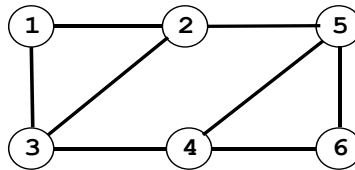
La vigilancia se ejerce recorriendo *circuitos de longitud* $k > 0$. Un circuito C es un conjunto de k puntos de control que pueden ser organizados en una secuencia $(a_0, a_1, \dots, a_{k-1})$, de manera que cada par $(a_i, a_{(i+1) \bmod k})$ es un sendero. Recorrer un circuito es lo mismo que visitar los puntos de control que lo conforman siguiendo la secuencia que lo define. Con esta secuencia se puede representar el camino, pero debe tenerse en cuenta que sus rotaciones, así como sus inversiones (escribir la secuencia en sentido contrario) representan el mismo circuito.

Problema

Dado un $k > 0$, la empresa quiere asignar un vigilante para recorrer, simultáneamente, cada circuito de longitud k . ¿Cuántos vigilantes debe tener disponibles?

Ejemplo:

Supóngase $V = 1 \dots 6$, con la siguiente configuración de segmentos:



Si $k=5$, la respuesta para el problema es 2, ya que hay 2 caminos circulares de longitud 5, a los que corresponden las secuencias

$(1, 2, 5, 4, 3)$
 $(2, 5, 6, 4, 3)$

Las secuencias

$(4, 3, 1, 2, 5)$
 $(4, 6, 5, 2, 3)$

representan los mismos caminos circulares arriba descritos.

Si $k=4$, solo se requiere un vigilante, ya que solo hay un circuito de longitud 4.

C Servicios

La empresa E recibe requerimientos de servicios para ser contratados en proyectos con otras empresas. Al momento de hacer la planeación de sus actividades, E tiene un total de n servicios requeridos, todos los cuales no tienen que ser necesariamente atendidos. En cambio, E realiza su planeación maximizando la ganancia que le pueden reportar algunos de los servicios requeridos.

Para fijar ideas, al momento de la planeación, E recibe $n > 0$ requerimientos de servicio. Un *servicio* se define con una cuádrupla (i, c, f, v) , donde

i : *identificador*
 c : día de *comienzo*
 f : día de *finalización*
 v : *valor*.

El identificador del servicio es un número en $1..n$. Los días c y f se definen como números naturales, $0 \leq c \leq f$. Los valores son números naturales positivos.

La empresa cuenta con $m > 0$ equipos de trabajo, cada uno de los cuales puede ser asignado a cualquier servicio. No es posible asignar más de un equipo a un servicio y, una vez que un equipo es asignado, no puede cambiarse esta decisión. Nótese que, si al momento de comienzo de un servicio no hay equipos disponibles para asignar, este servicio dejará de realizarse.

Problema

Determinar el máximo valor que E puede contratar.

Ejemplo:

Hay 7 servicios requeridos, descritos según la tabla siguiente:

| i | c | f | v |
|-----|-----|-----|-----|
| 1 | 3 | 4 | 20 |
| 2 | 1 | 1 | 50 |
| 3 | 2 | 99 | 200 |
| 4 | 6 | 18 | 100 |
| 5 | 1 | 2 | 30 |
| 6 | 2 | 3 | 50 |
| 7 | 10 | 10 | 70 |

Los requerimientos se pueden visualizar así:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | ... | 18 | ... | 99 |
|---|---|----|----|----|---|---|---|---|---|---|----|----|-----|----|-----|-----|-----|----|
| 1 | | | | 20 | | | | | | | | | | | | | | |
| 2 | | 50 | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | 200 | | |
| 4 | | | | | | | | | | | | | 100 | | | | | |
| 5 | | 30 | | | | | | | | | | | | | | | | |
| 6 | | | 50 | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | 70 | | | | | | | |

Si se tienen 2 equipos de trabajo, digamos, A y B, E puede maximizar su contrato de servicios obteniendo 400 unidades de valor, asignando los equipos de trabajo así:

A : 2, 4 y 6 (trabaja en los días 1, 2..3, 6..18)

B : 3 (trabaja en los días 2..99).

Si se tienen 3 equipos de trabajo, digamos, A, B y C, la contratación óptima es de 520, ya que todos los servicios se pueden contratar, asignando a A y B como en el caso anterior y a C así:

C : 1, 5 y 7 (trabaja en los días 1..2, 3..4, 10).

3 ENTRADA / SALIDA DE DATOS

En todas las soluciones que se presenten, la lectura de los datos de entrada se hace por la entrada estándar; así mismo, la escritura de los resultados se hace por la salida estándar.

Puede suponer que ninguna línea de entrada tiene espacios al principio o al final, y que los datos que se listan en cada línea están separados por exactamente un espacio.

A continuación, para cada problema, se establecen parámetros que definen su tamaño y formato de lectura de los datos, tanto de entrada como de salida.

3.1 Problema A: Longitud del A-segmento más largo

- Tamaño del problema: n
- Condiciones de los casos de prueba: $0 < n < 2 \cdot 10^4$.

Descripción de la entrada

La entrada contiene varios casos de prueba. Cada caso de prueba comienza con una línea de texto de la forma

n

donde $n > 0$ es el tamaño del arreglo de entrada $b[0 \dots n-1]$. Enseguida hay una línea con n números enteros

$b_0 \ b_1 \ \dots \ b_{n-1}$

donde b_i es el valor de $b[i]$, $0 \leq i < n$.

El fin de la entrada se indica con una línea que no se procesa, de la forma

0.

Descripción de la salida

Por cada caso de prueba para resolver, imprimir una línea de respuesta que contenga el tamaño del A-segmento más largo en el arreglo dado.

Ejemplos de entrada / salida

| Entrada | Salida |
|------------|--------|
| 1 | 1 |
| -3 | 3 |
| 5 | |
| 3 1 -1 2 0 | |
| 0 | |

3.2 Problema B: Vigilantes

- Tamaño del problema: n
- Condiciones de los casos de prueba: $0 < n < 5 \cdot 10^3$

Descripción de la entrada

La entrada contiene varios casos de prueba. Cada caso de prueba comienza con una línea de texto de la forma

$n \ k$

donde n y k son un números naturales positivos, que representan, respectivamente, el número de puntos de control y la longitud de los circuitos que los vigilantes deben cubrir.

A continuación, n líneas de la forma

1 $p_{11} \dots p_{1,r_1}$
2 $p_{21} \dots p_{2,r_2}$
...
 $n \ p_{n1} \dots p_{n,r_n}$

En la línea i , $1 \leq i \leq n$, $0 \leq r_i < n$, se representan senderos de la forma (i, p_{ij}) , con $1 \leq j \leq r_i$. Se garantiza que todos los elementos de la línea i son diferentes. Si hay un sendero (i, j) , esta información puede aparecer en la línea i , en la línea, j o en ambas (ya que los senderos se consideran iguales en las dos direcciones).

El fin de la entrada (un caso artificial que no se procesa) se indica con una línea de texto de la forma
0 0

Descripción de la salida

La salida debe tener exactamente una línea con un número entero que represente el número de vigilantes de los que la empresa debe disponer para satisfacer lo requerido

Ejemplo de entrada / salida

| Entrada | Salida |
|---|--------|
| 6 5 1 2 3 2 1 5 3 3 4 4 5 5 6 5 6 4 1 2 3 2 1 5 3 3 4 1 2 4 5 3 5 2 4 6 6 5 4 0 0 | 2 1 |

3.3 Problema C: Servicios

- Tamaño del problema: n
- Condiciones de los casos de prueba: $0 < n < 10^4$

Descripción de la entrada

La entrada contiene varios casos de prueba. Cada caso de prueba comienza con una línea de texto de la forma

$n \ m$

donde n es un número natural positivo, que determina el número de requerimientos de servicios que tiene E, y que hay m equipos de trabajo disponibles para cubrir los servicios que se realicen.

Enseguida hay n líneas con 4 números enteros. La i -ésima línea, $1 \leq i \leq n$ es de la forma

$$i \quad c \quad f \quad v$$

donde c, f son números naturales que representan el comienzo y el final del servicio i , y v es un número natural que representa su valor. Se tiene que $0 \leq c \leq f \leq 10^4$, $0 < v \leq 1000$.

El fin de la entrada se indica con una línea que no se procesa, de la forma

$$0 \quad 0$$

Descripción de la salida

La salida debe contener una línea con un número natural que representa al valor máximo contratable en el caso correspondiente.

Ejemplo de entrada / salida

| Entrada | Salida |
|------------|--------|
| 7 2 | 400 |
| 1 3 4 20 | 520 |
| 2 1 1 50 | |
| 3 2 99 200 | |
| 4 6 18 100 | |
| 5 1 2 30 | |
| 6 2 3 50 | |
| 7 10 10 70 | |
| 7 3 | |
| 1 3 4 20 | |
| 2 1 1 50 | |
| 3 2 99 200 | |
| 4 6 18 100 | |
| 5 1 2 30 | |
| 6 2 3 50 | |
| 7 10 10 70 | |
| 0 0 | |

4 ENTREGABLES

El proyecto puede desarrollarse por grupos de uno o dos estudiantes de la misma sección. La entrega se hace por Sicua+ (una sola entrega por grupo de trabajo).

El grupo debe entregar, por Sicua+, un archivo de nombre `proyectoDAlgo.zip`. Este archivo es una carpeta de nombre `proyectoDAlgo`, comprimida en formato `.zip`, dentro de la cual hay archivos fuente de soluciones propuestas y archivos que documentan cada una de las soluciones.

4.1 Archivos fuente de soluciones propuestas

Todos los programas implementados en *Java* deben compilar en *JDK 8*.

Para el problema X , siendo $X \in \{A, B, C\}$:

- Entregar un archivo *Java* (.java) con su código fuente de la solución que se presenta.
- Incluir como encabezado de cada archivo fuente un comentario que identifique el (los) autor(es) de la solución.
- Denominar `ProblemaX.java` el archivo de la solución que se presente.

Nótese que, si bien puede utilizarse un *IDE* como *Eclipse* durante el desarrollo del proyecto, la entrega requiere incluir solo un archivo .java por cada solución. El archivo debe poderse compilar y ejecutar independientemente (sin depender de una estructura de directorios, de un IDE, de librerías no estándar, etc.).

4.2 Archivos que documentan soluciones propuestas

Toda solución propuesta debe acompañarse de un archivo que la documente, con extensión .doc, .docx o .pdf. El nombre del archivo debe ser el mismo del código *Java* correspondiente. Por ejemplo, si incluyó un archivo `ProblemaB.java`, como solución para el problema B, debe incluirse un archivo `ProblemaB.docx` (o `ProblemaB.pdf`) que lo documente.

Un archivo de documentación debe contener los siguientes elementos:

- 0 *Identificación*
Nombre de autor(es)
Identificación de autor(es)
- 1 *Algoritmo de solución*
Explicación del algoritmo elegido. Si hubo alternativas de implantación diferentes, explicar por qué se escogió la que se implementó.
Deseable:
Anotación (contexto, pre-, poscondición, ...) para cada subrutina o método que se use.
- 2 *Análisis de complejidades espacial y temporal*
Cálculo de complejidades y explicación de las mismas. Debe realizarse un análisis para cada solución entregada.
- 3 *Comentarios finales*
Comentarios al desempeño observado de la solución.

Téngase en cuenta que los análisis de 2 tienen sentido en la medida que la explicación de 1 sea clara y correcta. No se está exigiendo formalismo a ultranza, pero sí que, como aplicación de lo estudiado en el curso, se pueda describir un algoritmo de manera correcta y comprensible.

No describa un algoritmo con código GCL a menos que lo considere necesario para explicarlo con claridad. Y, si lo hace, asegúrese de incluir aserciones explicativas, fáciles de leer y de comprender.