

TAREA 2

ALGORITMOS Y ESTRUCTURAS DE DATOS 1^{DO} SEMESTRE 2020

Objetivo de la tarea: Evaluar el entendimiento de colas y pilas.

Profesor: Karol Suchan

Ayudante: Yerko Ortiz

- Se aceptarán entregas de la tarea hasta el miércoles 8 de Abril a las 23:59 horas.
- Para la entrega se debe de subir un código en Java por cada problema vía classroom.
- Cada solución ha de estar en un único programa con extensión .java, donde el nombre del archivo debe incluir su nombre y apellido (evite usar espacios y caracteres raros en lo posible). (ejemplo: problema1Tarea2YerkoOrtiz.java)

Pilas Iguales

Usted tiene tres pilas de cilindros, donde todos los cilindros tienen el mismo diámetro pero pueden variar en altura; es posible cambiar la altura de una pila eliminando el cilindro de la parte superior las veces que sea necesario.

Diseñe e implemente un programa que encuentre la altura máxima de las pilas de forma tal que todas las pilas tengan la misma altura. Esto significa que usted debe remover cero o más cilindros de la parte superior de las pilas hasta que las tres tengan exactamente la misma altura. La eliminación de cilindros debe realizarse de forma tal que se **maximice** la altura final de las pilas.

Nota: una pila vacía sigue siendo una pila.

Entrada:

- La primera línea contiene tres enteros n_1 , n_2 , n_3 separados por espacios, que describen el respectivo número de cilindros en cada pila.
- La línea dos contiene n_1 enteros x_i que describen la altura de cada cilindro en la pila uno, de abajo hacia arriba.
- La línea tres contiene n_2 enteros x_j que describen la altura de cada cilindro en la pila dos, de abajo hacia arriba.
- La línea cuatro contiene n_3 enteros x_k que describen la altura de cada cilindro en la pila tres, de abajo hacia arriba.

Dominio de la entrada:

- $0 \leq n_1, n_2, n_3 \leq 10^5$
- $0 \leq x_i, x_j, x_k \leq 100$

Salida:

- Imprimir la altura máxima h , tal que las tres pilas tengan la misma altura h .

Nota: el ayudante les proveerá del programa que realiza la lectura de la entrada de datos, por lo que usted solo deberá enfocarse en resolver el problema (usted decide si usar el programa de lectura o hacer el suyo propio).

Caso de prueba:

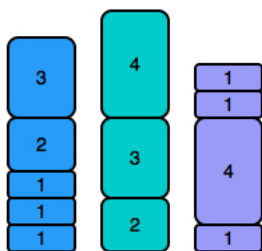
Entrada:

5	3	4
1	1	1
2	3	4
1	4	1

Salida:

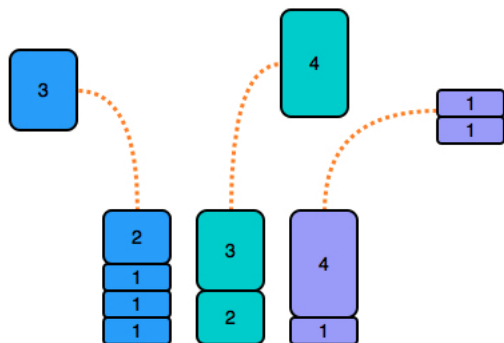
5

Explicación:



Estado inicial

Como es posible observar, inicialmente las tres pilas pueden tener distintas alturas. Para que las tres pilas queden del mismo tamaño es necesario remover el cilindro superior de la pila 2, luego el cilindro superior de la pila 1 y luego los dos cilindros superiores de la pila 3.



Estado final

Como resultado las tres pilas quedan del mismo tamaño:

$$9 - 4 = 5$$

$$8 - 3 = 5$$

$$7 - 2 = 5$$

Cola de impresión

Una impresora está experimentando una gran cantidad de peticiones para imprimir documentos, hay momentos donde llegan a haber hasta cientos de trabajos en la cola de impresión, por lo que para poder imprimir es necesario esperar horas.

Existen ciertos documentos que son más importantes que otros, por lo que el General Hacker ha inventado e implementado un sistema de prioridad para la impresora. Cada documento a imprimir es asignado a una prioridad entre 1 y 9, donde 9 es la prioridad máxima y 1 es la prioridad mínima; el sistema de prioridad de la impresora opera de la siguiente forma:

- El primer documento J es sacado de la cola.
- Si hay algún documento en la cola con una prioridad mayor que J en la cola, entonces J no se imprime y se inserta al final de la cola, a la espera de que se impriman los documentos con mayor prioridad.
- En caso contrario(que no exista un documento de mayor prioridad), el documento J es impreso(y no se vuelve a reinsertar al final de la cola).

De esta forma, todas las importantes recetas de pastelería que el General Hacker está imprimiendo, serán impresas de forma ágil y veloz. Donde por otro lado el resto de documentos, tendrán que esperar un rato hasta ser impresos según su prioridad.

El problema de esta nueva política es que se ha vuelto dificultoso determinar cuánto se debe de esperar hasta que un documento sea impreso. Es por esto que usted decide escribir un programa que realice este calculo. El programa va a simular la cola de impresión y calcular cuanto se debe esperar para que el documento x sea impreso. Para simplificar las cosas asuma que imprimir un documento siempre toma un minuto, y que añadir o eliminar documentos a la cola es instantáneo.

Entrada:

- Una línea con dos enteros n y m , donde n es el número de documentos en la cola y m es la posición del documento al cual se le calculará el tiempo de impresión. Las posiciones en la cola están enumeradas desde el 0 hasta el $n - 1$.
- La siguiente línea contiene n enteros x_i , que representan la prioridad de cada documento en la posición i de la cola.

Dominio:

- $1 \leq n \leq 100$
- $0 \leq m \leq n - 1$
- $1 \leq x_i \leq 9$

Nota: el ayudante les proveerá del programa que realiza la lectura de la entrada de datos, por lo que usted solo deberá enfocarse en resolver el problema (usted decide si usar el programa de lectura o hacer el suyo propio).

Salida:

- Imprimir un único entero con el número de minutos que el documento estuvo en la cola hasta ser completamente impreso.

Casos de prueba:

Entrada 1:

```
1 0
5
```

Entrada 2:

```
4 2
1 2 3 4
```

Entrada 3:

```
6 0
1 1 9 1 1 1
```

Salida 1:

```
1
```

Salida 2:

```
2
```

Salida 3:

```
5
```