Departamento de Computación
FCEFQyN, Universidad Nacional de Río Cuarto
Asignaturas: Diseño de Algoritmos - Algoritmos II
Primer Cuatrimestre de 2023

Trabajo Práctico No. 1

La resolución de este trabajo práctico debe ser completada a través de GitHub Classroom, antes de las **23:55 del sábado 6 de Mayo de 2023**.
El código provisto como parte de la solución a los ejercicios deberá estar documentado apropiadamente **usando Javadoc**. Aquellas soluciones que no requieran programación, como así también la documentación adicional de código que se desee proveer, debe entregarse en **archivos de texto convencionales**, en el mismo repositorio de entrega de las soluciones, con nombres que permitan identificar fácilmente su contenido.
Tanto la calidad de la solución, como el código y su documentación serán considerados en la calificación. Recuerde además que los trabajos prácticos **tienen defensa**, y que el trabajo en la resolución de los ejercicios debe realizarse de manera equitativa entre los miembros de cada grupo. Los grupos deben tener 3 integrantes.

– **Problema de clasificación de Mosquitos**.

Un grupo de investigación del laboratorio de microbiología necesita detectar el mosquito más raro de una muestra de mosquitos que fue recolectada siguiendo el mapa de casos positivos de dengue elaborado por el área de Salud.

Ingresaron al laboratorio N mosquitos, indexados de 0 a $N-1$. Cada mosquito tiene un tipo, que es un número entero entre 0 y 10^9 inclusive. Múltiples mosquitos pueden tener el mismo tipo.

Los mosquitos se pueden agrupadar por tipo. Definimos la cardinalidad del tipo de mosquito más frecuente como el cardinal más grande de los subgrupos de mosquitos. De manera similar, la cardinalidad del tipo de mosquito más raro es la cantidad de mosquitos del subgrupo más pequeño.

Por ejemplo, suponga que hay 11 mosquitos, cuyos tipos son $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. En este caso, la cardinalidad del tipo de mosquito más frecuente es 3. Los grupos con mayor número de mosquitos son el tipo 9 y el tipo 11, cada uno compuesto por 3 mosquitos. La cardinalidad del tipo de mosquito más raro es 1. Los grupos con la menor cantidad de mosquitos son el tipo 7, el tipo 0 y el tipo 100, cada uno de los cuales consta de 1 mosquito.

El laboratorio cuenta con una máquina con un solo botón que puede proporcionar información sobre los tipos de mosquitos. Inicialmente, la máquina está vacía. Para utilizar la máquina, se pueden realizar tres tipos de operaciones:

- Mover un mosquito al interior de la máquina.
- Mover un mosquito al exterior de la máquina.
- Pulsar el botón de la máquina.

Cada operación se puede realizar como máximo 40000 veces.

Cada vez que se presiona el botón, la máquina informa la cardinalidad del tipo de mosquito más frecuente, considerando únicamente los mosquitos dentro de la máquina.

Como parte de la solución a este trabajo práctico, se pide implementar, en Java, un programa que, dada una muestra de N mosquitos, determine la cardinalidad del tipo de mosquito más raro de la muestra **utilizando sólo** la máquina **como criterio para discriminar mosquitos**. El término **raro** hace referencia al tipo de mosquito menos frecuente.

Además, es importante tener en cuenta la cantidad máxima de cada operación que se realiza, ya que puede influir en la solución del problema (Revisar el perfil de cada método y los test asociados).

Detalles de implementacion

Debe implementar el siguiente método:

```
 /**
  * Return the cardinality of the rarest mosquito type.
  * @param sample the mosquito sample to analyze.
  * @return the cardinality of the rarest mosquito type.
  * @exception IllegalArgumentException if sample.length < 2 or sample.length > 2000.
  */
public static int rarest( int[] sample){

}


}
```

Este método debería devolver la cardinalidad del tipo de mosquito más raro entre los N mosquitos a analizar.

El método debe llamarse exactamente **una** vez.

El método rarest puede realizar llamadas a las siguientes operaciones de la máquina:

```
/**
 * Insert the ith element into the machine.
 * @param i the mosquito's index to be added.
 * @exception MachineException if the number of calls to this procedure exceeds the
     value 40000.
 * @exception IllegalArgumentException if i < 0 or i >= length of sample in the
     Machine.
 */
public void moveInside(int i);
```

Este procedimiento, inserta en la máquina el iésimo elemento pasado como parámetro. El valor de i debe estar entre 0 y $N-1$ inclusive.

Si este mosquito ya está dentro de la máquina, la llamada no tiene efecto sobre el conjunto de mosquitos en la máquina. Sin embargo, todavía se cuenta como una llamada separada.

Este procedimiento se puede llamar como máximo 40000 veces.

```
/**
 * Removes the ith element into the machine.
 * @param i mosquito's index to be eliminated.
 * @exception MachineException if the number of calls to this procedure exceeds the
     value 40000.
 * @exception IllegalArgumentException if i < 0 or i >= length of sample in the
     Machine.
 */
public void moveOutside(int i);
```

Este procedimiento, saca de la máquina el iésimo elemento pasado como parámetro. El valor de i debe estar entre 0 y $N-1$ inclusive.

Si este mosquito ya está fuera de la máquina, la llamada no tiene efecto sobre el conjunto de mosquitos en la máquina. Sin embargo, todavía se cuenta como una llamada separada.

Este procedimiento se puede llamar como máximo 40000 veces.

```
/**
 * Return the cardinality of the most frequent mosquito type.
 * considering only mosquitoes inside the machine.
```

```
 * @return the cardinality of the most frequent mosquito type.
 * @exception MachineException if the number of calls to this procedure exceeds the
     value 40000.
 */
public int pressButton();
```

Esta operación devuelve la cardinalidad del tipo de mosquito **más frecuente**, considerando únicamente los mosquitos dentro de la máquina.

Esta operació se puede llamar como máximo 40000 veces.

El clasificador no es adaptativo. Es decir, los tipos de los N mosquitos se fijan antes de llamar a **rarest** (Inicializar la población de mosquitos una sóla vez).

Ejemplo

Considere un escenario en el que hay una muestra de 6 mosquitos de los siguientes tipos $sample = [5, 8, 9, 5, 9, 9]$. El procedimiento rarest se llama de la siguiente manera:

```
int result = rarest(sample);
```

El procedimiento puede llamar a los métodos de la máquina, moveInside, moveOutside y pressButton de la siguiente manera:

Llamadas	Valor de retorno	Mosquitos en la máquina	Tipos de mosquitos en la máquina
		{}	[]
moveInside(0)		{0}	[5]
pressButton()	1	{0}	[5]
moveInside(1)		{0, 1}	[5, 8]
pressButton()	1	{0, 1}	[5, 8]
moveInside(3)		{0, 1, 3}	[5, 8, 5]
pressButton()	2	{0, 1, 3}	[5, 8, 5]
moveInside(2)		{0, 1, 2, 3}	[5, 8, 9, 5]
moveInside(4)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
moveInside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
pressButton()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
moveInside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
pressButton()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
moveOutside(5)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
pressButton()	2	{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]

En este punto, hay suficiente información para concluir que la cardinalidad del tipo de mosquito más raro es 1. Por lo tanto, el procedimiento rarest debería devolver 1.

En este ejemplo, moveInside se llama 7 veces, moveOutside se llama 1 vez y pressButton se llama 6 veces.

Restricciones

- $2 \leq N \leq 2000$
- Las operaciones moveInside, moveOutside y pressButton se pueden llamar, cada una, como máximo 40000 veces.

Si en cualquiera de los casos de prueba, las llamadas a los procedimientos moveInside, moveOutside o pressButton no se ajustan a las restricciones descritas en el enunciado, o el valor de retorno de rarest es incorrecto, influirá en la correción de la solución.

Algunas de las violaciones a las restricciones que serán evaluadas son las siguientes:

- **Parámetro no válido**: en una llamada a moveInside o moveOutside, el valor de i no está entre 0 y $N-1$ inclusive, o si la longitud de la muestra de mosquitos ingresada no está entre 2 y 2000 inclusive.
- **Demasiadas llamadas**: el número de llamadas a moveInside, moveOutside o pressButton supera 40000 veces.