

Laboratorio No. 1

Análisis de Algoritmos

(Tiempo de Ejecución, Algoritmos Recursivos)

Introducción

En este laboratorio el/la estudiante deberá determinar tiempos de ejecución en diferentes algoritmos. Además, deberá utilizar recursividad para resolver problemas que requieran soluciones recursivas

Objetivos

Al finalizar este laboratorio, el/la estudiante deberá ser capaz de:

- a. Determinar tiempos de ejecución para diferentes tipos de algoritmos
- b. Medir la eficiencia de un fragmento de código con respecto a otro (Unit Testing)
- c. Utilizar algoritmos recursivos para resolver problemas planteados
- d. Distinguir tipos de recursividad y aplicar soluciones funcionales a los problemas planteados
- e. Aplicar conocimientos discutidos en clase

Contexto

1. Trabaje con un modelo de n capas (domain, controller, testing, util).
2. Cree un nuevo proyecto llamado "Laboratory1" utilizando la tecnología javaFX, la cual permitirá trabajar en un entorno gráfico.
3. Defina una clase llamada "ThreeSum" e implemente el método `count(int a[])` publicado en el sitio de mediación del curso. Realice las pruebas requeridas utilizando el archivo `1Mints.txt` publicado en mediación.
4. Defina una clase llamada "Example" e implemente los métodos descritos en el libro Data Structures and Algorithms in Java, pág 183; publicados en mediación. Realice lo siguiente:
 - a. Cree una testing class (clase de testeo) llamada ExampleTest, utilizando la librería TestNG (testing library).
 - b. Utilice el siguiente fragmento de código para el análisis de cada método (example1, example2, example3, example4, example5):

```
Example ex = new Example();
int n = 10000000;
int a[] = new int[n];
fill(a); //llena el arreglo
```

```
//analisis de example1
long startTime = System.currentTimeMillis();
int result = ex.example1(a);
long endTime = System.currentTimeMillis();
long tn = endTime - startTime;
System.out.println("Test Example1\n"
+ "n: "+util.Utility.format(n)+"\n"
```

```
+ "Result: "+util.Utility.format(result)+"\n"
+ "Tn: "+util.Utility.format(tn)+ " milliseconds");
```

5. Defina una clase llamada "Recursive" e implemente los siguientes algoritmos en forma recursiva:

a. **Método:** double mcd(int x, int y)

Descripción: Algoritmo de Euclides que retorna el máximo común divisor entre dos números.

Uso: double result = mcd (x, y);

Gráficamente:

$$\text{mcd}(x, y) = \begin{cases} x & y = 0 \\ \text{mcd}(y, x \% y) & y > 0 \end{cases}$$

b. **Método:** int ackerman(int m, int n)

Descripción: Algoritmo de Ackerman que toma dos números naturales como argumentos y devuelve un único número natural.

Uso: int result = ackerman(m, n);

Gráficamente:

$$A(m, n) = \begin{cases} n+1 & m = 0 \\ A(m-1, 1) & m > 0, n = 0 \\ A(m-1, A(m, n-1)) & m > 0, n > 0 \end{cases}$$

c. **Método:** int min(int a[], int index, int value)

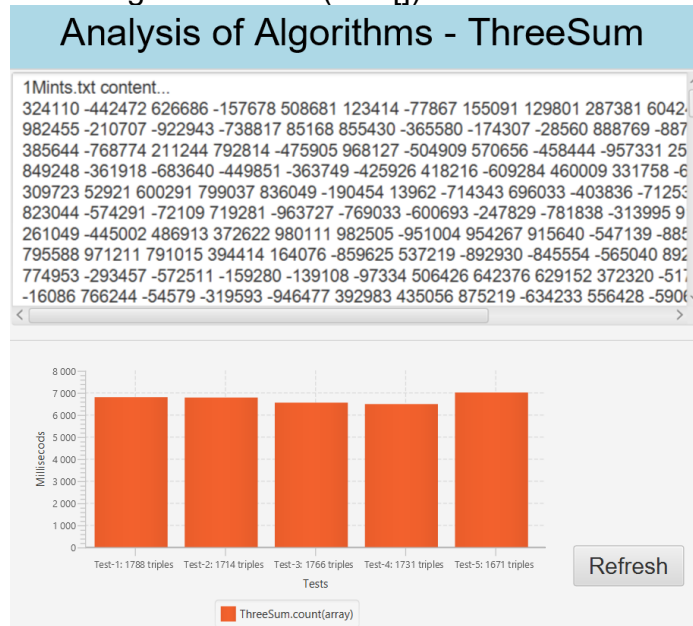
Descripción: Devuelve el elemento menor de un arreglo de números enteros

Uso: int result = min(a, 1, a[0]);

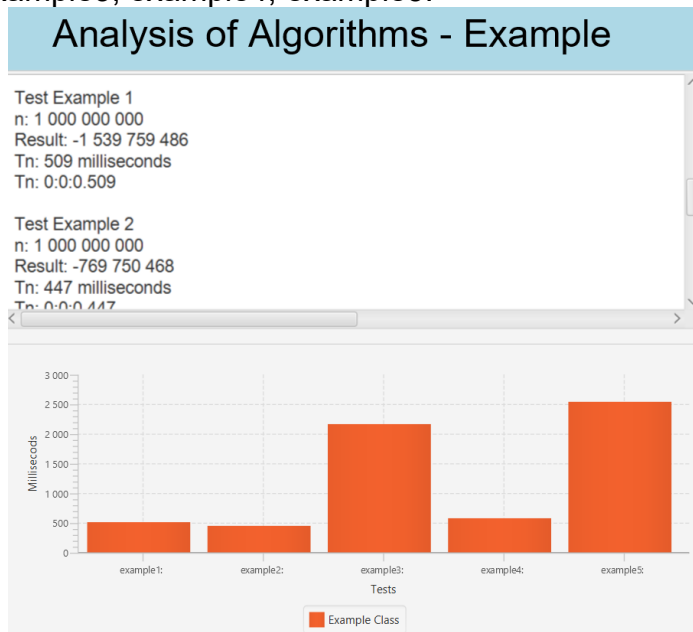
6. Compruebe el funcionamiento de la clase "Recursive" a través de una "testing class", de la siguiente forma:

- Encuentre el máximo común divisor para los siguientes pares de números: {(3, 5), (2, 4), (6, 8), (3, 6), (4, 8), (10, 14)}.
- Obtenga el tiempo de ejecución total del algoritmo.
- Pruebe la función de Ackerman con valores entre 0 y 3.
- Defina y llene un arreglo con 100 números enteros
- Muestre el contenido del arreglo por consola
- Utilice el método "min" para obtener el elemento menor del arreglo.
- Obtenga el tiempo de ejecución total del algoritmo.

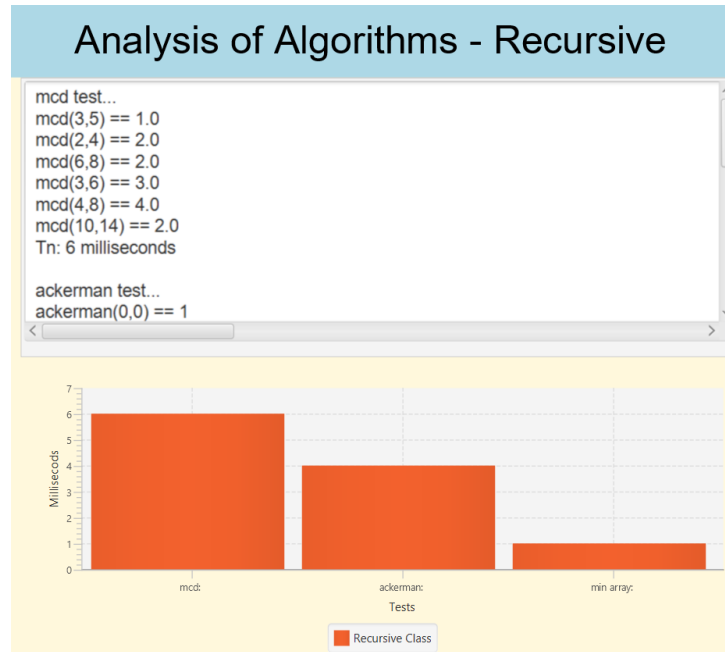
7. Utilice la tecnología “javaFX” para crear un entorno gráfico que muestre un menú principal y la solución a los algoritmos anteriores, de la siguiente forma:
- a. ThreeSum: Utilice un objeto gráfico tipo TextArea para mostrar el contenido del archivo de texto, y un objeto gráfico tipo BarChart para graficar los test ejecutados sobre el algoritmo count(int a[]).



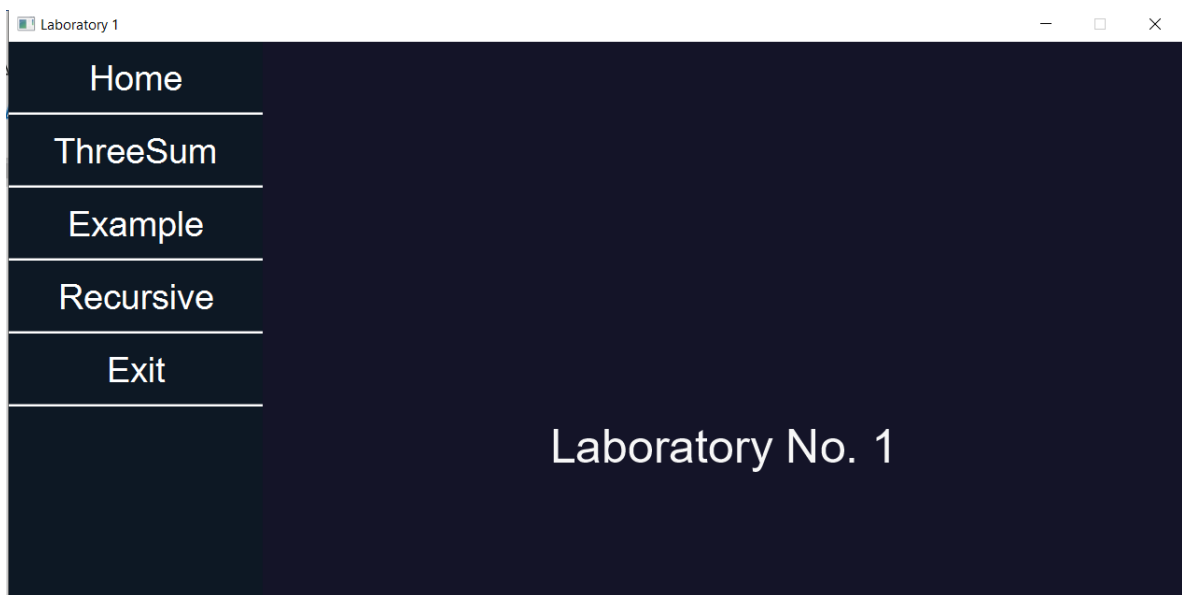
- b. Example: Similar al caso anterior, utilizando los algoritmos example1, example2, example3, example4, example5.



- c. Recursive: Similar a los anteriores, utilizando los algoritmos de la clase Recursive.



8. Un ejemplo del menú gráfico es el siguiente:



Resuelva y publique el laboratorio en el entorno del curso de la plataforma de mediación virtual (METICS). Verifique la fecha límite para el envío del informe.

URL: <https://mv1.mediacionvirtual.ucr.ac.cr/course/view.php?id=7513>