Laboratorio #1

Introducción

En este laboratorio se espera que El/la Estudiante utilice conceptos revisados en clase para solucionar problemas sencillos utilizando algoritmos recursivos, Además de forma experimental hacer revisión y análisis de algoritmos dados.

Objetivos

1. Poner en práctica conocimientos revisados en clase.
2. Utilización de algoritmos recursivos para resolver problemas planteados (directos e indirectos).
3. Evaluar algoritmos y su tiempo de ejecución.
4. Mostrar de forma grafica resultados de algoritmos en tiempo de ejecución (se busca la resolución de un problema a partir una investigación rápida e implementación de una librería nueva). http://www.jfree.org/jfreechart/

Ejercicios

1. Trabaje en los siguientes ejercicios utilizando recursividad
   a. Crear un algoritmo recursivo para determinar si un numero es par utilizando recursividad **indirecta**.
   El usuario debe ingresar los valores, retornar un mensaje amigable.
   Ejemplo: entrada 5 salida false, entrada 4 salida true

   b. Crear un algoritmo recursivo que reciba un string e indique si conforma una palabra **palíndroma**, En caso que la entrada sea una letra indicara que no es palíndroma.
   *"**Ejemplo** de **Palíndromos**. Se entienden como palíndromo o **palíndromos**, a las **palabras** o frases que se leen de igual manera hacia adelante o hacia atrás."*
   Ejemplo: Ana, oso, somarramos, sometemos

   c. Crear un algoritmo **iterativo** que retorne el factorial de un número dado. Además crear una solución recursiva

   Ejemplo entrada 4 salida 24

   4!=4*3*2*1=24

   d. Crear un método recursivo que reciba un arreglo de números enteros y lo ordene de forma descendente y retorne el arreglo ordenado.

Nota: Estos ejercicios deben entregarse con al menos dos pruebas unitarias cada uno.

2. Según los algoritmos proporcionados:
   Realice un análisis experimental de tiempo de ejecución, con al menos 5 muestras considerables el programa debe realizar las pruebas y devolver una salida con los resultados de forma tabulada

Muestra	Algoritmo1	Algoritmo2

```java
public static String repeat1 (char c, int n) {
    String answer = "";
    for (int j = 0; j < n; j++)
        answer += c;
    return answer;
}
public static String repeat2 (char c, int n) {
    StringBuilder sb = new StringBuilder ();
    for (int j = 0; j < n; j++)
        sb.append (c);
    return sb.toString();;
}
```

3. Dado el siguiente código:
   - Determine que hace el algoritmo
   - Determine las operaciones primitivas
   - Determine el Big O
   - Grafique comportamiento donde se refleje el peor comportamiento que podemos obtener

```java
public static boolean unknown(int numero) {
    int contador = 2;
    boolean result = true;
    while ((result) && (contador != numero)) {
        if (numero % contador == 0) {
            result = false;
        }
        contador++;
    }
    return result;
}
```

```java
public static int unknown(int n, int arr[]){
    for(int i = 0; i < n; i++)
        if(i < 2)
            arr[i] = 1;
        else
            arr[i] = arr[i-1] + arr[i-2];
    return arr[n-1];
  }
```