

Laboratorio Nro. 3

Listas enlazadas y Vectores dinámicos

Santiago Ochoa Castaño
Universidad Eafit
Medellín, Colombia
sochoac1@eafit.edu.co

Miguel Ángel Zapata Jiménez
Universidad Eafit
Medellín, Colombia
mazapataj@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Estructura de datos con el mapa de la ciudad de Medellín

Nota: Insertar en un arreglo dinámico, tiene complejidad $O(n)$

```
public static void readData(String nameFile) throws FileNotFoundException{
    String namefile; // C1
    if(nameFile.endsWith(".txt")){
        namefile=nameFile; // C2
    } else{
        namefile= nameFile + ".txt"; // C3
    }
    Scanner read = new Scanner( new File(namefile)); //C4
    double cont = 0; // C5
    while(read.hasNextLine()){ // (C6)n
        String line = read.nextLine(); // (C7)n
        Scanner scan = new Scanner(line); // (C8)n
        String a = scan.next(); / (C7)n
        double id = Double.parseDouble(a); // (C9)n
        a = scan.next(); / (C7)n
        double x = Double.parseDouble(a); // (C10)n
        a = scan.next(); / (C7)n
        double y = Double.parseDouble(a); // (C11)n
        if(scan.hasNext()){
            a = scan.next(); // (C12)n
            while(scan.hasNext()){ // C13 / Nota: Este while se debe a que hay direcciones que
                tienen entre dos o tres palabras por lo que no se considera un ciclo que se repite n veces. El peor de los
                casos seria cuando es tres. De acuerdo a la información leída a las pruebas de la carpeta dataset.
                a = a + " " + scan.next();
            }
        }else{
            a = null; // (C14)n
        }
        Ubicacion cord = new Ubicacion(id, x, y, a); // (C15)n
        mapa.add(cord); // (C16)n2
```

ESTRUCTURA DE DATOS 1

Código ST0245

```

        cont++; // (C17)n
    }
    read.close(); // C18
    System.out.println("Se han leído" + cont + " ubicaciones"); // C19
}

```

Complejidad Asintótica:

$$T(n) = C_1 + C_2 + C_3 + C_4 + C_5 + (C_6)n + (C_7)n + (C_8)n + (C_9)n + (C_{10})n + (C_{11})n + (C_{12})n + C_{13} + (C_{14})n + (C_{15})n + (C_{16})n^2 + (C_{17})n + C_{18} + C_{19}$$

$$O(C_1 + C_2 + C_3 + C_4 + C_5 + (C_6)n + (C_7)n + (C_8)n + (C_9)n + (C_{10})n + (C_{11})n + (C_{12})n + C_{13} + (C_{14})n + (C_{15})n + (C_{16})n^2 + (C_{17})n + C_{18} + C_{19})$$

$$O(n + n^2)$$

$$O(n^2)$$

La complejidad está dada por la cantidad de elementos añadir al arreglo

3.3 Teclado Roto algoritmo

R/ En este algoritmo se aprecia que la complejidad en tiempo de ejecución es $O(n)$

```

public class TecladoRoto
{
    public static void solucion(String s){
        s = separarEspacios(s); // (C1)n
        Nota: Este método se implementó para separar la cadena por espacios.
        Stack<String> lista = new Stack<String>(); //C2
        String[] n = s.split(" "); //C3
        String textoIni = ""; //C4
        String textoFin = ""; //C5
        String textoLista = ""; //C6
        String texto = ""; //C7
        for(int i = 0; i < n.length; i++){ //C8 + (C9)n
            if(n[i].equals("[") && i+1 < n.length){ //C10)n
                if(!n[i+1].equals("]") && !n[i+1].equals("[")){ //C11)n
                    textoIni = n[i+1] + textoIni; //C12)n
                    i++; //C13)n
                }
            }else if(n[i].equals("]")){ // (C14)n
                if(!n[i+1].equals("[") && !n[i+1].equals("]")){ //C15)n
                    textoFin = textoFin + n[i+1]; //C16)n
                    i++; // (C17)n
                }
            }else if(!n[i].equals("[") && !n[i].equals("]")){ //C18)n
                lista.push(n[i]); //C19)n
            }
        }
    }
}

```

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
 Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

Código ST0245

```

while(!lista.isEmpty()){//(C20)n
    textoLista = textoLista + lista.pop(); //(C21)n
}
texto = textoIni + textoLista + textoFin; // C22
System.out.println(texto); // C23
}

```

Complejidad Asintótica:

$$T(n) = (C_1)n + C_2 + C_3 + C_4 + C_5 + C_6 + (C_7)n + (C_8)n + (C_9)n + (C_{10})n + (C_{11})n + (C_{12})n + (C_{13})n + (C_{14})n + (C_{15})n + (C_{16})n + (C_{17})n + (C_{18})n + (C_{19})n + (C_{20})n + (C_{21})n + C_{22} + C_{23}$$

$$O((C_1)n + C_2 + C_3 + C_4 + C_5 + C_6 + (C_7)n + (C_8)n + (C_9)n + (C_{10})n + (C_{11})n + (C_{12})n + (C_{13})n + (C_{14})n + (C_{15})n + (C_{16})n + (C_{17})n + (C_{18})n + (C_{19})n + (C_{20})n + (C_{21})n + C_{22} + C_{23})$$

O(n)

3.4 Explicación de las variables de cálculo presentes en la complejidad del ejercicio 3.3

La complejidad está dada por el tamaño de la cadena tipo "String" enviada como parámetro al método solución.

4) Simulacro de Parcial

4.1 b) y **4.1.2 a)**

4.2 c)

4.4 stack.pop() **4.4.1 a)**

4.5 a)

4.6 b)

4.8 d) y **4.8.1 b)** y **4.8.2 c)** y **4.8.3 c)**

4.9.1 d) y **4.9.2 a)** y **4.9.3 b)**

4.10.1 b) y **4.10.2 d)**

4.11.1 s1.size() > 0 y **4.11.2 s1.pop()** y **4.11.3 s2**

4.12.1 iv) y **4.12.2 i)**

4.13.1 i) y **4.13.2 i)**

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas

Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627

Tel: (+57) (4) 261 95 00 Ext. 9473