

## **Laboratorio Nro. 3**

# **Listas enlazadas y listas hechas con arreglos.**

**Juan Andrés Giraldo Aristizábal**

Universidad Eafit

Medellín, Colombia

jagiraldoa@eafit.edu.co

**Camilo Cossio Alzate**

Universidad Eafit

Medellín, Colombia

ccossioa@eafit.edu.co

### **3) Simulacro de preguntas de sustentación de Proyectos**

#### **3.1**

<b>Ejercicio</b>	<b>Complejidad con ArrayList</b>	<b>Complejidad con LinkedList</b>
<b>1.1</b>	<b>O(n^2)</b>	<b>O(n^2)</b>
<b>2.1</b>	<b>O(n^3logn)</b>	<b>O(nlogn)</b>

La diferencia en eficiencia de acuerdo con la complejidad se empieza a notar cuando se trata de insertar elementos al inicio o al final como en el ejercicio 2.1

**3.2** Primero se crea un LinkedList con el propósito de usar las clases addFirst y addLast, ya que con estas se hace más fácil el trabajo de concatenar al inicio o al final. Creamos un String auxiliar que más adelante permitirá crear el string que irá al inicio o el final de la oración de acuerdo con los corchetes. Después se hace un ciclo que se encarga de buscar si hay corchetes, si hay se ingresa a otro ciclo que se encarga de crear aux, es decir, la palabra que esta entre dos corchetes para ponerla al inicio o final de la oración. Luego, mediante dos condicionales, se determina si aux debe ir al inicio o al final de la linkedList, esto se da dependiendo del corchete, si es abierto va al inicio y si es cerrado, al final. Después se inicializa la variable aux en vacío para repetir el proceso anteriormente descrito. Si hay palabras que no están dentro de ningún corchete, simplemente se añade al linkedList. Finalmente, se imprime cada posición del linkedList.

**3.3** La complejidad del ejercicio 2.1 es  $O(nlogn)$

**3.4** La n representa la longitud del string que entra como parámetro

### **4) Simulacro de Parcial**

**4.1 A.**

**4.2 C.**

**4.3** A)  $q.size() > 1$   
B) <

**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

- C) q.add(q.remove());
- D) q.remove();
- 4.4** A) lista.size()
- B) lista.addLast(auxiliar.pop())
- 4.5.1** auxiliar1.size() > 0  
auxiliar2.size() > 0
- 4.5.2** personas.offer(edad);
- 4.6** B.
- 4.8** O(n).
- 4.8.1** A.
- 4.8.2** C.
- 4.8.3** C.
- 4.9.1** D.
- 4.9.2** A.
- 4.9.3** B.
- 4.10.1** C.
- 4.10.2** A.
- 4.11.1** s1.size() > 1
- 4.11.2** s1.pop()
- 4.11.3** s1.pop
- 4.12.1** iv
- 4.12.2** i
- 4.13.1** iv.
- 4.13.2** ii
- 4.14** iii

**5) Lectura recomendada (opcional)**

Mapa conceptual

**6) Trabajo en Equipo y Progreso Gradual (Opcional)**

**6.1** Actas de reunión

**6.2** El reporte de cambios en el código

**6.3** El reporte de cambios del informe de laboratorio

**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

