**Laboratorio Nro. 3  
Listas enlazadas y listas hechas con arreglos**

|  |  |
| --- | --- |
| **Manuela Herrera López**  Universidad Eafit  Medellín, Colombia  mherreral@eafit.edu.co | **Samuel Palacios Bernate**  Universidad Eafit  Medellín, Colombia  sdpalaciob@eafit.edu.co |

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

**3.1.**

|  |  |  |
| --- | --- | --- |
|  | **ArrayList** | **Linkedist** |
| **1.1** | **Buscar en O(n)**  **Get en O(1)** | **Buscar en O(n)**  **Get en O(n/4)** |

Es preferible usar un ArrayList cuando no se tiene mucha memoria, y es mejor usar un LinkedList cuando se quieren añadir elementos en una posición determinada

**3.2** El ejercicio 2.1 (teclado roto) funciona de la siguiente forma; primero que nada, se lee el input, y luego se evalúa carácter por carácter para ver si alguno de estos es un ‘[’ o un ‘]’. Si es alguno de estos modificamos una variable que lo que hace es decirnos dónde se va a agregar el texto que sigue; posteriormente, carácter por carácter se van agregando a la lista enlazada una posición después de la variable mencionada anteriormente.

**3.3** la complejidad del algoritmo es de O(n)

**3.4** n en el caso de la complejidad es la cantidad de caracteres que tiene la entrada

***4) Simulacro de Parcial***

* 1. *b, insertar un carácter en cualquier posición*
  2. *b, O(1)*
  3. *a. while(q != null)*

b. for(int i =1; i<num; i++)

c. q.siguiente

d. p.remove()

4.4. a. lista.size >0

b. lista.add(auxiliar.pull())

4.5 línea 12: auxiliar1.size >0

Línea 16: auxiliar2.size >0

Línea 18: personas.pull()

4.6. b O(n2)

4.7. c O(n)

4.8. a. O(k)

4.8.2 c. retorna 12

4.8.3. c. O(1)

4.9.1 d. O(n)

4.9.2 a. retorna 6

4.9.3. b O(n)

4.10.1. a O(n2)

4.10.2. d O(1)

4.11.1 while (s1.size > 0)

4.11.2 s1.pop()

4.11.3 s2

4.12.1 imprime d

4.12.2 a O(1)

4.13.1 c O(n2)

4.13.2 d O(n3)

4.14 c. 2, 3, 4, 5