

Taller 1 AREP

santiago martínez

January 2021

1 Introducción

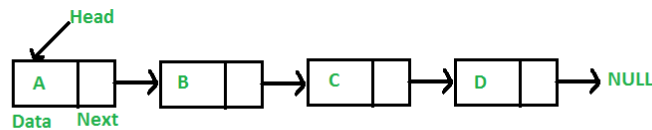
En este documento explicaremos como implementamos nuestra propia linkedlist y que métodos asociamos a ella para un buen funcionamiento, con esto, resolver la necesidad de poder con nuestra propia linkedlist leer una serie de datos de tipo Double y con esto obtener la media y su desviación estándar.

2 Linked List

Una Linked List o lista enlazada es una estructura de datos fundamental y puede ser usada para implementar otras estructuras de datos. Su estructura es simple, son una serie de nodos enlazados, los cuales tienen un valor y tienen una referencia al nodo siguiente de la lista, por eso se le llama de tipo autor referenciado porque contiene un puntero o enlace a otro dato del mismo tipo(Nodo).

3 Mi implementación

Mi implementación se basa en dos clases principalmente, la clase LinkedList y la clase Nodo, con la clase LinkedList implemento las funcionalidades que necesito para poder trabajar con esta lista custom, para esto implemente 3 métodos, los cuales son, add (el cual añade un nuevo nodo a la lista), size(el cual retorna la longitud de la lista) y get(el cual retorna el valor de un nodo dependiendo de la posición que sea ingresada).



4 Metodo add

Este método tiene un parámetro de entrada, el cual es el valor que va a contener el nodo, este método evalúa si la lista esta previamente vacía, si es así el parámetro first, el cual hace referencia al primer nodo dejara de ser null y ahora se referenciara a este nuevo nodo, en el caso de que no sea el primero, se evaluará si el ultimo nodo esta vacío, con esto tenemos encuentra la segunda inserción y se le asignara como el nodo enlazado del nodo first.

```
public void add(Double value) {
    Node newNode = new Node(value);
    if (first == null) {
        first = newNode;
    } else {
        if (last == null) {
            first.setNextNode(newNode);
            last = newNode;
        } else {
            last.setNextNode(newNode);
            last = newNode;
        }
    }
}
```

5 metodo get

Este método es de suma importancia ya que gracias a este logramos de alguna forma recorrer la lista, para implementamos, lo que hicimos fue tener un parámetro de entrada el cual nos indica la posición en la lista que deseamos, y luego, vamos recorriendo cada nodo hasta que lleguemos a la posición deseada. el recorrido se da gracias al nodo siguiente, que en este caso, es el nodo

```

public Double get(int index) {
    if (first != null) {
        if (index == 0) {
            return first.getValue();
        } else {
            int i = 0;
            Node next = first.getNextNode();
            Double value = null;
            while (i != index) {
                i++;
                value = next.getValue();
                next = next.getNextNode();
            }
            return value;
        }
    } else {
        return null;
    }
}

```

6 Conclusiones

Con este taller nos dimos cuenta la importancia de saber implementar estructuras de datos para poder solucionar de manera eficiente los requerimientos que nos son pedidos y con esto ser eficientes y ágiles a la hora de resolver problemas utilizando la programación.

References

- [1] *Linked List Data Structure.*
<https://www.geeksforgeeks.org/data-structures/linked-list/>
- [2] *Data Structure and Algorithms - Linked List.*
https://www.tutorialspoint.com/data_structures_algorithms/linked_list_algorithms.htm
- [3] textitLinked Lists.
<https://www.interviewbit.com/courses/programming/topics/linked-lists/>