What is the Singly Linked List (SLL)?

In this lesson, we will be introduced to the Singly Linked List and its structure, along with some basic operations that this data structure offers.

We'll cover the following

Introduction

Linked Lists and **Arrays** are quite similar in characteristics as they are both used to store a collection of the same type of data. A data type can be anywhere from a simple primitive integer to a complex object of a particular class. However, the structure of the Linked List is very different as compared to Arrays.

Applications

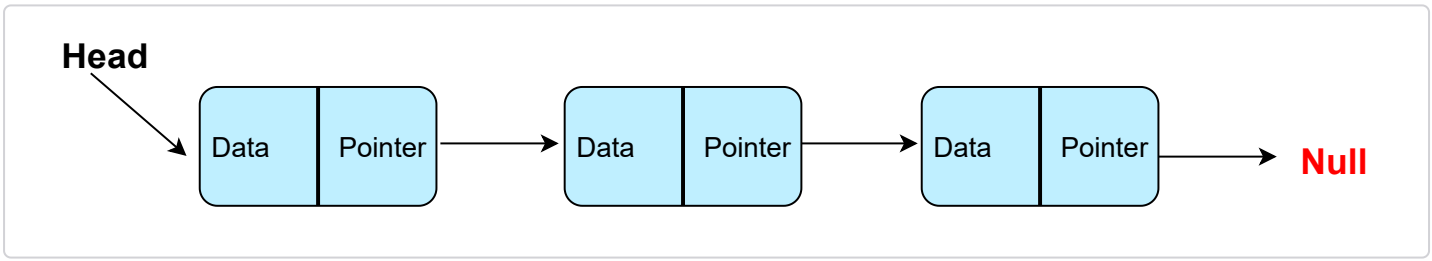
Some important applications of Linked Lists include:

- Implementing HashMaps, File System and Adjacency Lists
- Dynamic memory allocation: We use linked lists of free blocks
- Performing arithmetic operations on long integers
- Maintaining a directory of names

Structure of Linked List

A linked list is formed by *nodes* that are linked together like a chain. Each node holds data, along with a pointer to the next node in the list. The **Singly Linked List** (SLL) is the type of linked list where each node has only one pointer that stores the

reference to the next value. The following illustration shows a Singly Linked List.



In the next few lessons, we will look at how SLL is implemented and study its basic operations in detail. To implement a linked list, we need the following two classes:

- Class Node
- Class LinkedList

Class Node

The `Node` class stores data in a single node. It can store primitive data such as *integers* and *string* as well as complex objects having multiple attributes. Along with data, it also stores a pointer to the next element in the list, which helps in linking the nodes together like a chain. Here's a typical definition of a `Node` class:

```
1  //Class node having Generic data-type <T>
2  public class Node<T> {
3    public T data; //Data to store (could be int, string, Object etc)
4    public Node nextNode; //Pointer to next node in list
5  }
```

Node Class

Explanation: The code above provides a basic definition of a *Node* class structure with two data members: one to hold the data and the other one to store a reference to the next element. This class is a generic class made by using Java Generics (https://docs.oracle.com/javase/tutorial/java/generics/index.html). The `data` variable can hold any data-type value specified.

Class Linked list

As mentioned above, the Singly Linked list is made up of nodes that are linked together like a chain. Now to access this chain, we would need a pointer that keeps track of the first element of the list. As long as we have information about the first element, we can traverse the rest of the list without worrying about memorizing

their storage locations.

The Singly Linked List contains a `head` node: a pointer pointing to the first element of the list. Whenever we want to traverse the list, we can do so by using this head node. Below is a basic structure of the Singly Linked List's class:

```
public class SinglyLinkedList<T> {
    //Node inner class for SLL
    public class Node{
        public T data; //Data to store (could be int, string, Object etc)
        public Node nextNode; //Pointer to next node in list

    }

    public Node headNode; //head node of the linked list
    public int size;      //size of the list

    //constructor
    public SinglyLinkedList() {
        headNode = null;
        size = 0;
    }
}
```

Singly Linked List Class

Explanation: In the above class, we see a `headNode` variable of type `Node`. It will point to the start of the list (i.e., head of the list). In the constructor, we initialize that `headNode` node so that it becomes functional. If a head node points to nothing (`NULL`), this means that the list is *empty*. As you can see in the above illustration, the head points to the first element and the last pointer points to *NULL* to indicate the end of the list.

Next, we'll look at all the methods which are present in a standard `LinkedList` class.

Report an Issue

Ask a Question (https://discuss.educative.io/tag/what-is-the-singly-linked-list-sll__linked-lists__data-structures-for-coding-interviews-in-java)