Linear Data Structures.

Ortiz Vega Angelo
Course Code: CE1103
Name: Algorithms and Data Structures I,
Academic Area of Computer Engineering.
Cartago, Costa Rica.

-Keywords: Linear Data Structures, Queues, Linked List, Stacks, Arrays.

-Content:

Linear data structures are a type of abstract data (ADT). Mathematical Concept: Definition of types and associated set of operations. Class?

Common ADT:

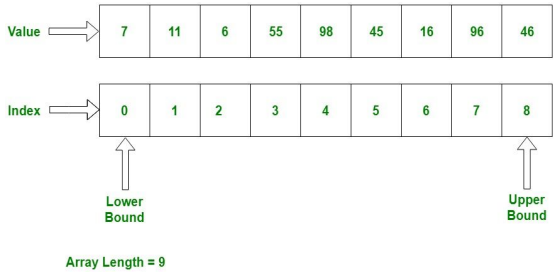
~ Linked Lists.
~ Batteries.
~ Tails.
~ Trees
~ Graphs.

Most languages include their own ADT implementations.

Arrays:

The most basic data structure supported by most languages are arrays. An array is a contiguous block of memory that can accommodate several elements of the same type. Accessing elements of an array is extremely fast. Equivalent to make a sum.

They can not grow or decrease with ease, since when creating it, the desired size is defined. Adding items in the middle of the array is expensive.



Simple Lists:

It is a structure composed of nodes linked together. Each node is an independent memory object with respect to the others.

They are dynamic, means that the number of nodes is not fixed can grow or decrease. Each node is an object that has at least two attributes, Value: stored data; Next: reference to the next node.

Simple Example: Summary

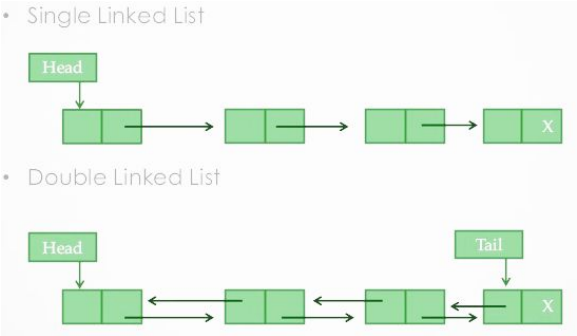
```
import java.util.*;

public class SimpleList {
  public static void main(String[] args) {
    List<String> words = new ArrayList<>();
    words.add("hi");
    words.add("hello");
    words.add("hola");
    System.out.println("First word: " + words.get(0));
    System.out.println("All words: " + words);
  }
}
```

```
First word: hi
All words: [hi, hello, hola]
```

Double Linked Lists

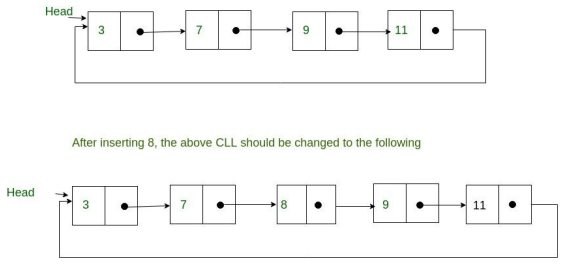
They are a variation of simple lists. The advantage is the backward navigability. Each node has a reference to the next node and a reference to the previous node. The disadvantage is that two references must be handled when inserting or deleting the list.



Circular Lists:

They allow browsing the entire list from any node.

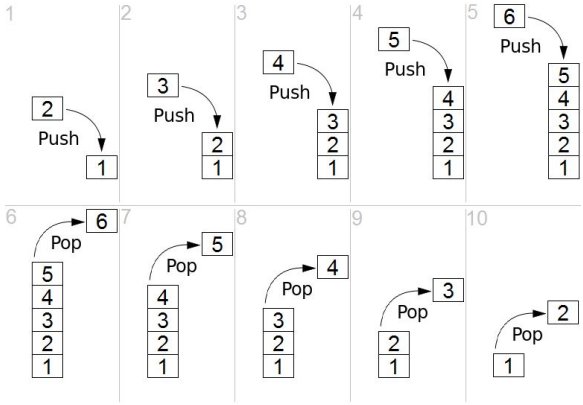
Every node has a successor. There is no concept of first or last, you only know where you enter the list. If the list is ordered it is easier to perform operations.



Stack:

ADT with nature LIFO (Last In, First Out).They only allow access to the element at the top. Supports operations such as:

1. Push: Insert elements in the top.
2. Pop: Remove element on the top.
3. Peek: Get the element in the top without removing it.



Queues:

Similar to a stack but with FIFO behavior (First In, First Out) It allows to abstract many situations from real life. Supports 3 operations:

1. Enqueue: Insert at the end.
2. Dequeue: Take to the Start.
3. Peek: Take the item at the beginning without removing it.