**Important Data Structures and their respective API's**

1. Arrays <-> ArrayList

2. Linked List <-> Stack and Queue

3. Binary Tree <-> Binary Heap

**Importance of Arrays:**

1. Arrays can contain anything

Examples:

Object[] obj = new Object[10];

String[] str= new String[10];

Student[] st= new Student[10];

1. Have Fixed Size -> cannot grow
2. Can randomly access i.e. its Big O is O(1) i.e. directly fetch the data from a specific index

**Importance of Linked Lists:**

1. Can grow, nothing specific to size
2. Insertion is easy.
3. Stack and Queue are the implementations available
4. Understanding the concept is important, once ur clear, it will be easy to think and write the code
5. It has a concept **of Node which contains data and pointer to next node**

Insertion

* At beginning is O(1)
* In between is O(1) until n(-> n is the index)
* At end is O(n) as we need to iterate through the last node