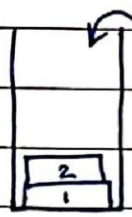
URBAN EDGE

Introduction to Stack Data Structure

Stack is a linear data structure. Operations on stack are performed in LIFO (Last in first out) order.



⇒ Item 2 which entered the basket last will be the first one to come out

LIFO (Last in first out)

Applications of Stack

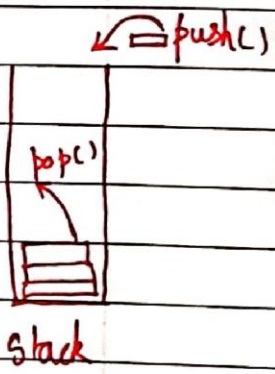
1. Used in function calls
2. Infix to postfix conversion (and other similar conversions)
3. Parenthesis matching & more...

Stack ADT

In order to create a stack we need a pointer to the topmost element along with other elements which are stored inside the stack.

Some of the operations of Stack ADT are:

1. push() → push an element into the stack
2. pop() → remove the topmost element from the stack
3. peek(index) → Value at a given position is returned
4. isEmpty() / isFull() → Determine whether the stack is empty or full.



Implementation

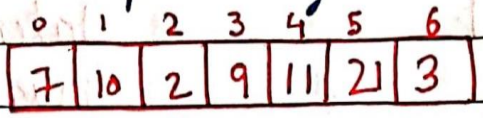
A stack is a collection of elements with certain operations following LIFO (Last in First out) dicipline.

A Stack can be implemented using an array or a linked list

Linear Vs Binary Search

Linear Search

Searches for an element by visiting all the elements sequentially until the element is found.

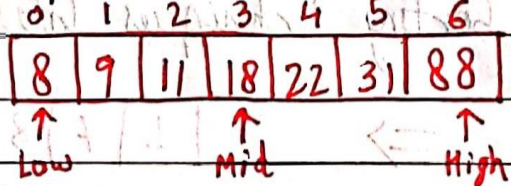


⇒ can be sorted or unsorted

Search '2' → Element found WC Complexity : O(n)

Binary Search

Searches for an element by breaking the search space into half in a sorted array.



WC Complexity O(log n)

Search 18

The search continues towards either side of mid based on whether the element to be searched is lesser or greater than mid.

	Linear Search	Binary Search
1.	Works on both sorted and unsorted arrays	Works only on sorted arrays
2.	Equality operations	inequality operations
3.	O(n) WC complexity	O(log n) Wc Complexity