Stack

Additional document

Stack

This is another type of linear data structure that is useful in various situations. This data structure follows the property of LIFO (Last In First Out). A real-life example of this data structure you can observe when you perform undo operation on your system. You will be using Linked List and Array to implement this data structure.

You can consider stack as a collection of donuts that are stored on top of each other. Now at any point in time, you can access the donut which is at the top.

In this segment, you will be implementing the following problem statements.

1. **Initialization and declaration:** This problem statement will work on the initialization and declaration part of Stack. Here we will insert a few elements in the stack and then print them accordingly.
2. **Push operation:** This is the classical terminology used instead of insert. Push operation takes O(1) time in order to maintain the behavior of stack. Here you will be inserting an element by using a variable called top.
3. **Pop operation:** This is another terminology used only in the stack, this term signifies data deletion in the stack. Here, the last inserted item will be deleted first from the stack. So in order to access or delete the first element, it will take O(n) time.
4. **Isempty:** This algorithm will check if the given stack currently has some elements in it or not. If it is empty it will return true else False.
5. **Peek:** This algorithm will peek into the stack and will print the last inserted value from the stack. This algorithm will take O(1) time to perform.