Stack Applications

Stack is a simple linear data structure used for storing data. Stack follows the LIFO(Last In First Out) strategy that states that the element that is inserted last will come out first. You can take a pile of plates kept on top of each other as a real-life example. The plate which we put last is on the top and since we remove the plate that is at the top, we can say that the plate that was put last comes out first. It can be implemented through an array or linked lists. Some of its main operations are: **push(), pop(), top(), isEmpty(), size(), etc.** In order to make manipulations in a stack, there are certain operations provided to us. When we want to insert an element into the stack the operation is known as the push operation whereas when we want to remove an element from the stack the operation is known as the pop operation. If we try to pop from an empty stack then it is known as underflow and if we try to push an element in a stack that is already full, then it is known as overflow.

Primary Stack Operations:

- void push(int data): When this operation is performed, an element is inserted into the stack.
- int pop(): When this operation is performed, an element is removed from the top of the stack and is returned.

Auxiliary Stack Operations:

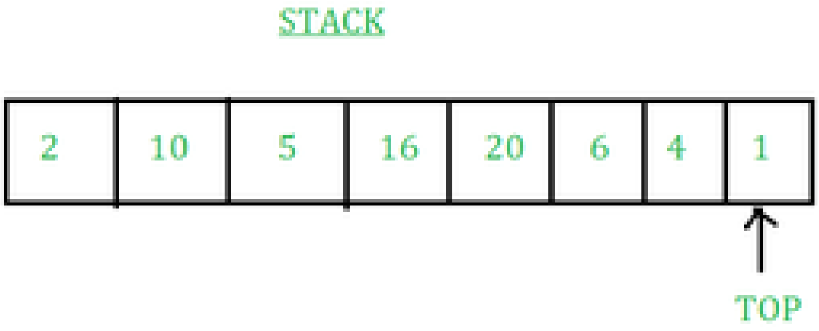
- **int top():** This operation will return the last inserted element that is at the top without removing it.
- **int size():** This operation will return the size of the stack i.e. the total number of elements present in the stack.
- **int isEmpty():** This operation indicates whether the stack is empty or not.
- **int isFull():** This operation indicates whether the stack is full or not.

Types of Stacks:

- Register Stack: This type of stack is also a memory element present in the memory unit and can handle a small amount of data only. The height of the register stack is always limited as the size of the register stack is very small compared to the memory.
- Memory Stack: This type of stack can handle a large amount of memory data. The height of the memory stack is flexible as it occupies a large amount of memory data.

What is meant by Top of the Stack?

The pointer through which the elements are accessed, inserted, and deleted in the stack is called the **top of the stack**. It is the pointer to the topmost element of the stack.



Application of Stack Data Structure:

- Stack is used for evaluating expression with operands and operations.
- Matching tags in HTML and XML
- Undo function in any text editor.
- Infix to Postfix conversion.
- Stacks are used for backtracking and parenthesis matching.
- Stacks are used for conversion of one arithmetic notation to another arithmetic notation.
- Stacks are useful for function calls, storing the activation records and deleting them after returning from the function. It is very useful in processing the function calls.
- Stacks help in reversing any set of data or strings.

Application of Stack in real life:

- CD/DVD stand.
- Stack of books in a book shop.
- Undo and Redo mechanism in text editors.
- The history of a web browser is stored in the form of a stack.
- Call logs, E-mails, and Google photos in any gallery are also stored in form of a stack.
- YouTube downloads and Notifications are also shown in LIFO format(the latest appears first).

Advantages of Stack:

- Stack helps in managing data that follows the LIFO technique.
- Stacks are be used for systematic Memory Management.
- It is used in many virtual machines like JVM.
- When a function is called, the local variables and other function parameters are stored in the stack and automatically destroyed once returned from the function. Hence, efficient function management.
- Stacks are more secure and reliable as they do not get corrupted easily.
- Stack allows control over memory allocation and deallocation.
- Stack cleans up the objects automatically.

Disadvantages of Stack:

- Stack memory is of limited size.
- The total of size of the stack must be defined before.
- If too many objects are created then it can lead to stack overflow.
- Random accessing is not possible in stack.
- If the stack falls outside the memory it can lead to abnormal termination.