



Basic Operations on Stack using Data Structure

Mr. A. S. Salvi Kumbhar, Ms. Pranali Bedkyale, Ms. Pranali Kore, Ms. Sakshi Killedar

Lecturer, DKTE, DKTE's Yeshawantrao Chavan Polytechnic Ichalkaranji, Ichalkaranji-416115, India

Student, DKTE, DKTE's Yeshawantrao Chavan Polytechnic Ichalkaranji, Ichalkaranji-416115, India

ABSTRACT

In computer science, a stack is an abstract data type that serves as a collection of elements, with two main operations: Push, which adds an element to the collection, and Pop, which removes the most recently added element that was not yet removed. Calculators employing reverse Polish notation use a stack structure to hold values. Expressions can be represented in prefix, postfix or infix notations and conversion from one form to another may be accomplished using a stack. Many compilers use a stack for parsing the syntax of expressions, program blocks etc. before translating into low-level code. Most programming languages are context-free languages, allowing them to be parsed with stack-based machines.

Keywords: push, pop, insert, delete

1. Introduction

Stacks in Data Structures is a linear type of data structure that follows the LIFO (Last-InFirstOut) principle and allows insertion and deletion operations from one end of the stack data structure, that is top. Implementation of the stack can be done by contiguous memory which is an array, and non-contiguous memory which is a linked list. Stack plays a vital role in many applications. The stack data structure is a linear data structure that accompanies a principle known as LIFO (Last In First Out) or FILO (First In Last Out). Real-life examples of a stack are a deck of cards, piles of books, piles of money, and many more.

1.1 Example of stack

This example allows you to perform operations from one end only, like when you insert and remove new books from the top of the stack. It means insertion and deletion in the stack data structure can be done only from the top of the stack. You can access only the top of the stack at any given point in time.

- Inserting a new element in the stack is termed a push operation.
- Removing or deleting elements from the stack is termed pop operation.

1.2 Stack representation in data structure

A stack can be implemented by means of Array, Structure, Pointer, and Linked List. Stack can either be a fixed size one or it may have a sense of dynamic resizing. Here, we are going to implement stack using arrays, which makes it a fixed size stack implementation. stack represented by two ways:

1. Array Representation of Stacks

First we have to allocate a memory block of sufficient size to accommodate the full capacity of the stack. Then, starting from the first location of the memory block, the items of the stack can be stored in a sequential fashion. In Figure, Item_i denotes the ith item in the stack; l and u denote the index range of the array in use; usually the values of these indices are 1 and SIZE respectively. TOP is a pointer to point the position of the array up to which it is filled with the items of the stack.

2. Linked List Representation of Stacks

Although array representation of stacks is very easy and convenient but it allows the representation of only fixed sized stacks. In several applications, the size of the stack may vary during program execution. An obvious solution to this problem is to represent a stack using a linked list. A single linked list structure is sufficient to represent any stack. Here, the DATA field is for the ITEM, and the LINK field is, as usual, to point to the next' item. Above Figure b depicts such a stack using a single linked list. 2 In the linked list representation, the first node on the list is the current item that is the item at the top of the stack and the last node is the node containing the bottom-most item. Thus, a PUSH operation will add a new node in the front and a POP operation will remove a node from the front of the list.

1.3 Push Operations

Push operation involves inserting new elements in the stack. Since you have only one end to insert a unique element on top of the stack, it inserts the new element at the top of the stack.

Logic operation of push –

```
Void push ()
{ if(top==max-1)
  {
    Printf("stack is full");
  }
  else{
    Stack[top]=value;
    Top=top+1;
  } }
```

1.4 Pop Operation

Pop operation refers to removing the element from the stack again since you have only one end to do all top of the stack. So removing an element from the top of the stack is termed pop operation. File naming and delivery

Logic of pop operation –

```
Void pop()
{ If(top==1)
  {
    Printf(" stack is empty ")
  }
  Else {
    Value=stack[top];
    Top= top-1; } }
```

2. Illustrations

The following are the basic operations served by stacks.

- push: Adds an element to the top of the stack.
- pop: Removes the topmost element from the stack.
- isEmpty: Checks whether the stack is empty.
- isFull: Checks whether the stack is full.
- top: Displays the topmost element of the stack.

3. Algorithm

Algorithm for push operation

PUSH(stack, Top , Max, Item)

This algorithm push on element on the stack.

Step 1 : If TOP=MAX-1 then

Write "OVERFLOW" and return.

Step 2 : Set TOP = TOP +1

Step 3 : Set STACK[TOP]= ITEM

Step 4: Exit

Algorithm for pop operation

POP(stack, Top , Item)

This algorithm POP on element from stack.

Step 1 : If TOP=-1 then

Write "UNDERFLOW" and return.

Step 2 : Set ITEM_STACK[TOP]

Step 3 : Set TOP=TOP-1

Step 4 : Exit

Acknowledgements

I would like to convey my heartfelt gratitude to Mr. A. P. Salvi Kumbhar for her tremendous support and assistance in the completion of my project. I would also like to thank our Principal, Mr. A. P. Kothali, for providing me with this wonderful opportunity to work on a project with the topic Basic operation on stack. The completion of the project would not have been possible without their help and insights..

References

[https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

<https://www.scaler.com/topics/stack-operations-in-data-structures/>

<https://www.geeksforgeeks.org/introduction-to-stack-data-structure-and-algorithm-tutorials/>

<https://www.geeksforgeeks.org/basic-operations-in-stack-data-structure-with-implementations/>