

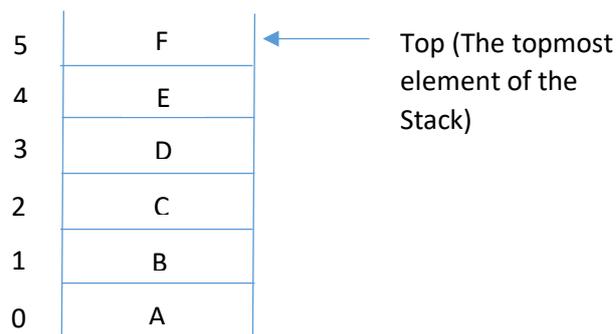
Data Structure

Stack

Definition :

Stack is a linear data structure, an ordered list of elements, in which items can be entered into and retrieved from only one end, called **Top**. Stack is also called Last In First Out (LIFO) list as the element entered last into the Stack has to be retrieved first. We can show it by the following diagram-

Elements- A, B, C, D, E, F....



The element F entered last into the above stack but it would be the first element that will leave the stack, hence stack is LIFO list.

The Top holds the topmost element of the stack. Top starts from -1. Then each time an element enters into the stack Top will be incremented first then new element goes into the stack at the position of Top. Each time an element leaves the stack is pointed by Top and after its departure Top is decremented.

In the above stack if new element G comes then Top will first be incremented and then G goes to stack. In other way if F leaves stack then Top will point E as topmost element of the stack.

But here two very important points arises if Top already reaches the extent of the stack then what happen? Stack will be overflowed if there is a new element to be entered. Hence this condition must be monitored each time an element enters into the stack. Another case if the stack is already empty then an attempt to retrieve an element means stack underflow. So, both these condition checking is must at the time insertion and deletion from the stack.

Stack as Data Structure –

Stack is a linear data structure, also an user-defined data structure that can be implemented through Array and also through Linked list. Stack can be of any domain, it depends on the type of the data stored in the Stack. The main functions performed on a Stack are as follows-

- 1) Create() : Create a stack and initialize the Top
- 2) PUSH() : Enter a new element into the Stack.
- 3) POP() : Retrieve or delete the topmost element of the Stack
- 4) TOP() : Returns the topmost element of the stack does not delete it.
- 5) Is_empty() : Checks whether the stack empty or not. Return 1 if empty, else return 0.

Axioms:

Stack S[] is global data structure used by all the above functions, Top is also global, can be modified by any function, PUSH takes as input the new element, POP will return the deleted element etc. are the axioms or rules....

Algorithms-

- 1) Procedure Create()
 1. Start
 2. Top is set to -1
 3. End of the procedure

- 2) Procedure PUSH(ne : integer)
 1. Start
 2. If Top reaches the MAXSIZE of Stack S then print "Stack Overflow"
 3. Else begin
 4. Top will be incremented by 1.
 5. Store the new element ne into S[Top]
 6. end
 7. End of the procedure

- 3) Procedure POP() : integer
 1. Start
 2. If Top = -1 then print "Stack Underflow"
 3. Else begin
 4. Store the topmost element S[Top] into pe. //Popped
 5. Top will be decremented by 1.
 6. End
 7. Return pe
 8. End of the procedure

- 4) Procedure TOP() : integer
 1. Start
 2. If Top = -1 then print "Stack Underflow"
 3. Else begin
 4. Return the topmost element S[Top]
 5. End
 6. End of the procedure

- 5) Procedure Is_empty() : integer
 1. Start
 2. If Top = -1 then return 1
 3. Else return 0.
 4. End
 5. End of the procedure

Note : Algorithms are given for array. Try for linked list also.