Stack

Arnab Ganguly, Assistant Professor
Department of Computer Science, University of Wisconsin – Whitewater
Data Structures (CS 223)

1 Stack

A stack is a collection of items (integers, char, float, etc.) which allows the following two operations:

- *push*(i) : adds an item i to the collection, and
- *pop*() : returns and removes the LATEST item, provided the stack is not empty

Besides the above two operations, a stack may also support the following additional operations:

- *peek*() : returns the latest item from the collection without removing it,
- *size*() : returns the number of items in the stack, and

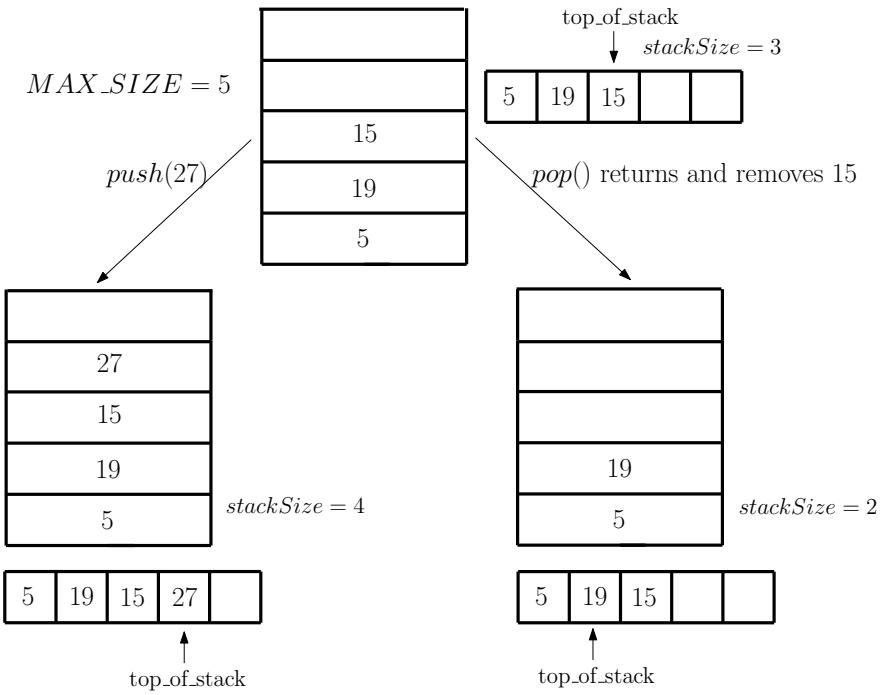

Figure 1: Stack: array that simulates it, *top_of_stack*, and MAX_SIZE variables

Algorithm 1 Implementation of a Stack

```
1: int MAX_SIZE = 5, top_of_stack = −1;
2: int stackArray[MAX_SIZE]; // an array which simulates the stack
3:
4: function PUSH(int val)
5:     if (top_of_stack == MAX_SIZE − 1) then
6:         "Cannot push! Stack is full";
7:     else
8:         stackArray[++top_of_stack] = val;
9:
10: function POP()
11:     if (top_of_stack == −1) then
12:         "Cannot pop! Stack is empty";
13:     else
14:         return stackArray[top_of_stack−−];
15:
16: function PEEK()
17:     if (top_of_stack == −1) then
18:         "Cannot peek! Stack is empty";
19:     else
20:         return stackArray[top_of_stack];
21:
22: function SIZE()
23:     return (top_of_stack + 1);
```