DATA STRUCTURES AND ALGORITHM

Linked Stack

By
Zainab Malik

Content

- Limitation of Stack using Arrays
- Representation of Stack using a Linked List

Limitation of Arrays

Array based implementation of abstract data structures like stack suffer from following limitations.

1. Size of the Stack must be known in advance
2. We may come across a situation where an attempt to push an element causes overflow.
3. Array based representation prohibits the growth of the stack beyond the finite numbers of elements

Solution: Linked List

Linked List representation allows stack to grow to a limit of the computer's available (free) memory.

Limitation of Array based Stack

Stack

Operation	Top	0	1	2	3	4	5
Push(g)	5	a	b	c	d	e	f

Error: Stack overflow

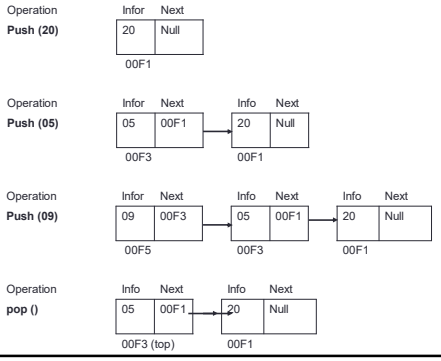
5

Solution:

Linked List Representation of Stack
or
Linked Stack

6

Linked List Representation of Stack



7

Linked List Representation of Stack

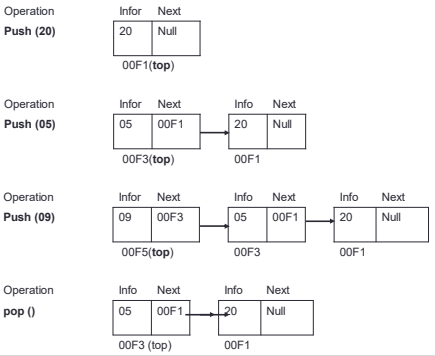
- Just like a singular linked list, in linked stack, each node has two slots
  - First slot contains the information/content
  - Second slot contains the pointer/address of the next node

Info	Next
12	Null

- Instead of head and tail pointers, in a linked stack there is only one pointer "top" that contains the address of first node always.
  - When there is no element in the linked stack, top contains null
  - To ensure FILO or LIFO order
    - The push operation is implemented by inserting a new node at the start of the list (**AddAtBeginning()**)
    - The pop operation is implemented by removing the node from the start of the list (**RemoveFromBeginning()**)

8

Linked List Representation of Stack



9

Operation of Linked Stack

- **push(item):** We may implement function same as addAtBeginning() function of singular linkedlist (AI-Lecture03)
- **pop ():** We may implement function same as removeFromBeginning() function of singular linked list (AI-Lecture03)
- **isEmpty():** Need to check top pointer, if contains null or 0 means linked stack is empty
- **isFull():** No need to implement because there is no fixed size in linked list representation of stack
- **topValue():** Need to return top->info
- **removeAll():** this is also known as destructor in which we delete all nodes one by one till top becomes 0
  - Which function to call for the deletion of these nodes?

10

Thank You