TECHNO INDIA UNIVERSITY
WESTBENGAL

Data Structures & Algorithms
(PCC-CS 301)

Dr. Debashis Das
Associate Professor
Department of CSE
Techno India University, Kolkata

Topics Covered

1. Linear Data Structure
   a. Array
   b. Stack

Array: Data Structure

- Array
  - Property
    - It is defined as a sequential storage of similar type data
    - It is a linear data structure
    - Elements are stored in each cell of the structure one by one
  - Representation
    - Each element is accessed through its cell index

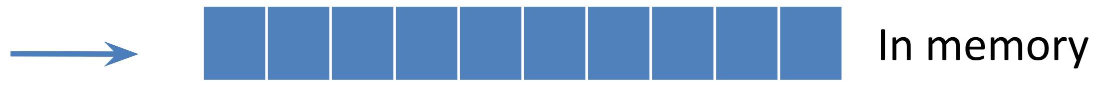
A	10	20	30	40	50	60	70	80	90	100
	0	1	2	3	4	5	6	7	8	9

A[4] = 50 , A[0] = 10

Array: Data Structure

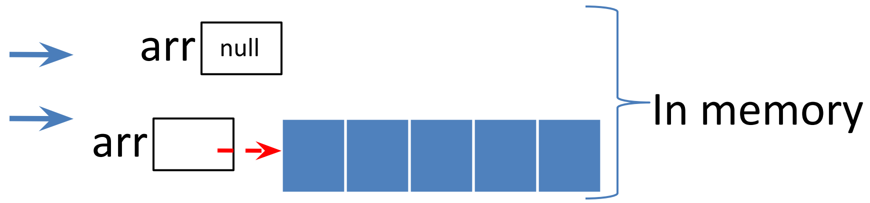
- Array
  - Declaration
    - Using static memory allocation:
      - array is declared along with its size

```
int arr[10];
```



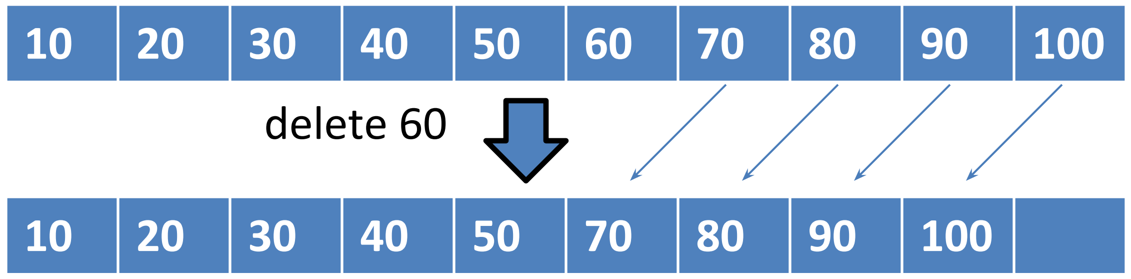
    - Using dynamic memory allocation
      - Array size can be defined on demand instead of fixing it at the time of declaration
      - Implemented by using the concept of pointer

```
int  *arr;
arr= (int*) malloc(n*sizeof(int));
```



Array: Data Structure

- Array
  - Operations
    - Data insertion
      – Data is inserted into each cell starting from the beginning
    - Data deletion
      – Data deletion can be performed in any arbitrary position
      – It is performed by shifting of all the next elements by one position towards left (no intermediate cell should be vacant)

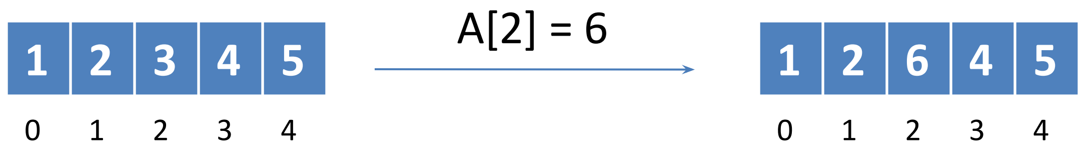


Array: Data Structure

- Array
  - Operations
    - Data display
      - All the elements are displayed one at a time from the beginning

```
For  i = 1 to n
    print A[i]
```

    - Data searching
      - Searching of a specific element in the array **(will be covered later)**
    - Data modification
      - Replacement of new data in any specific cell directly



Array: Data Structure

- Array Operation: complexity

Operation	Time Complexity
Data insertion (single data)	O(1)
Data deletion	O(n)
Display array	O(n)
Data modification	O(1)
* Data insertion (entire array)	O(n)

Array: Data Structure

- Array
  - Advantage
    - Easy to implement and access
  - Disadvantage
    - Cannot deal with multiple types of data, only similar type data can be stored in an array
    - Requires sequential memory space to store the entire array
    - It is not a suitable data structure for storing a large number of data
    - Data deletion is time consuming (cost effective)

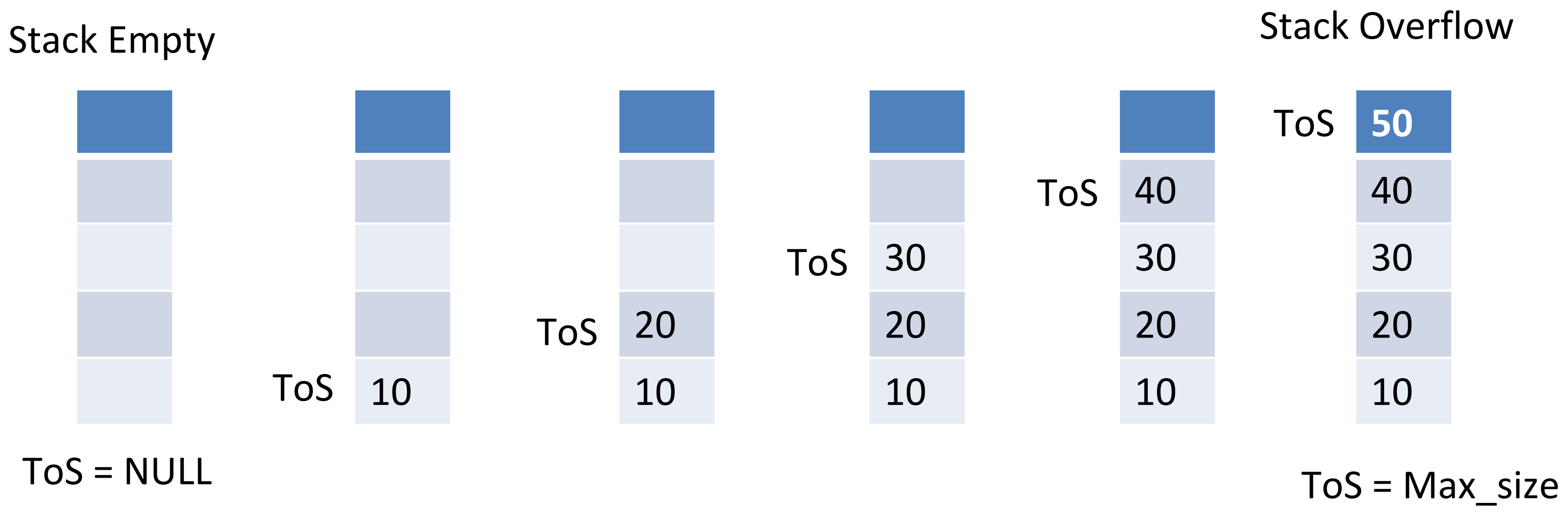
TECHNO INDIA UNIVERSITY
WESTBENGAL

Stack: Abstract Data Type

- Stack
    - Properties
        - Stack is defined as a Last In First Out (LIFO) data structure
            - The last data inserted into the stack to be deleted first
        - The associated operations of a stack are also defined with the data structure that is why it is considered as an ADT
        - The top most element of the stack is pointed by Top of Stack (ToS) pointer
        - ToS will hold NULL for an empty stack
        - All the stack operations will be performed through ToS
        - Stack size is fixed which should be defined at the beginning

Stack: Abstract Data Type

- Stack
  - Representation



Stack: Abstract Data Type

- Stack
  - Operations
    - PUSH (data insertion into stack)
    - POP (data deletion from stack)
    - Display (showing element of stack)

    Primary operation

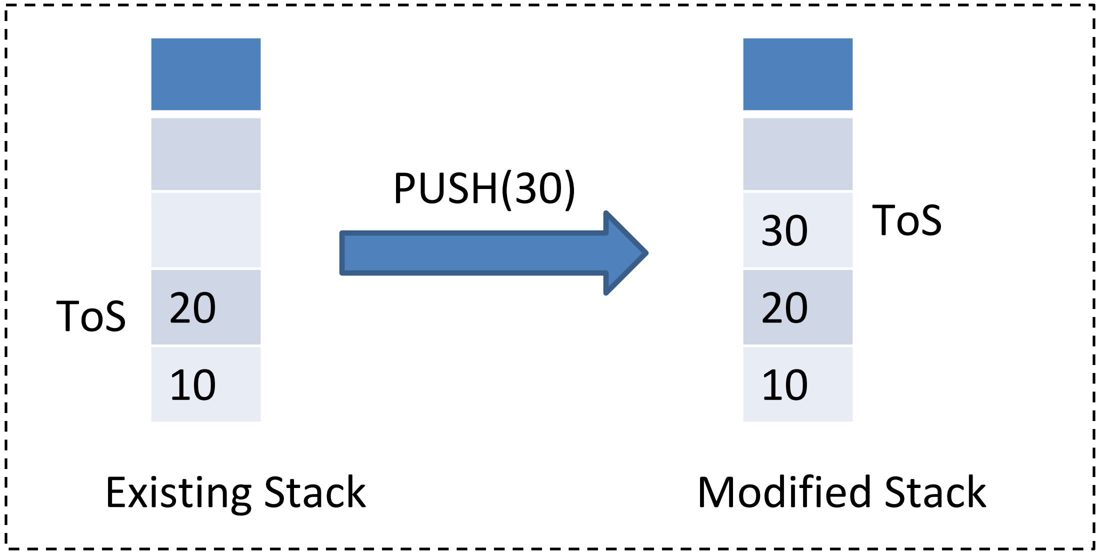
    - IsFullStack (checks if stack is overflow)
    - IsEmptyStack (checks if stack is underflow)

    Auxiliary operation

Stack: Abstract Data Type

- Stack Operation
  - PUSH
    - This function inserts one element at the top most position of the stack if the stack is not full
    - The newly inserted data is pointed by ToS

```
void PUSH(element)
{
  if  IsFullStack = TRUE
    return
  else
    tos := tos+1
    Stack(tos) := element
}
```

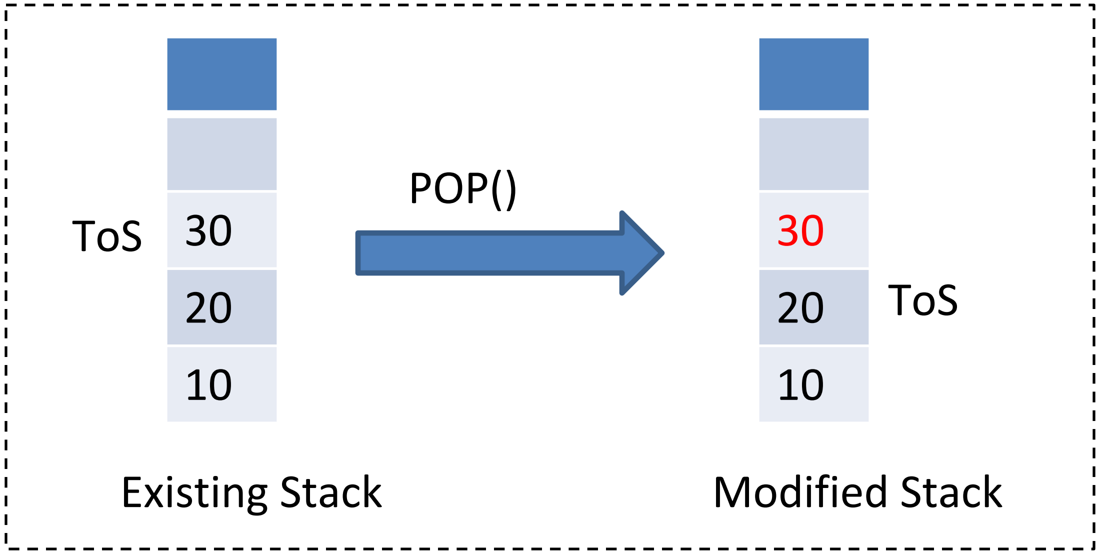


TECHNO INDIA UNIVERSITY
WESTBENGAL

Stack: Abstract Data Type

- Stack Operation
  - POP
    - This operation deletes the top most element of the stack if it is not empty
    - The current top most element will be pointed by ToS

```
int POP()
{
 if  IsEmptyStack = TRUE
   return NULL
 else
   data := Stack(tos)
   tos := tos-1
   return data
}
```



Queries?