

Data Structure

Tasmina Akter

October 10, 2018

Data structure

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way.

Data Structures is about rendering data elements in terms of some relationship, for better organization and storage.

For example, we have some data which has, player's name "Virat" and age 26. Here "Virat" is of String data

type and 26 is of integer data type.

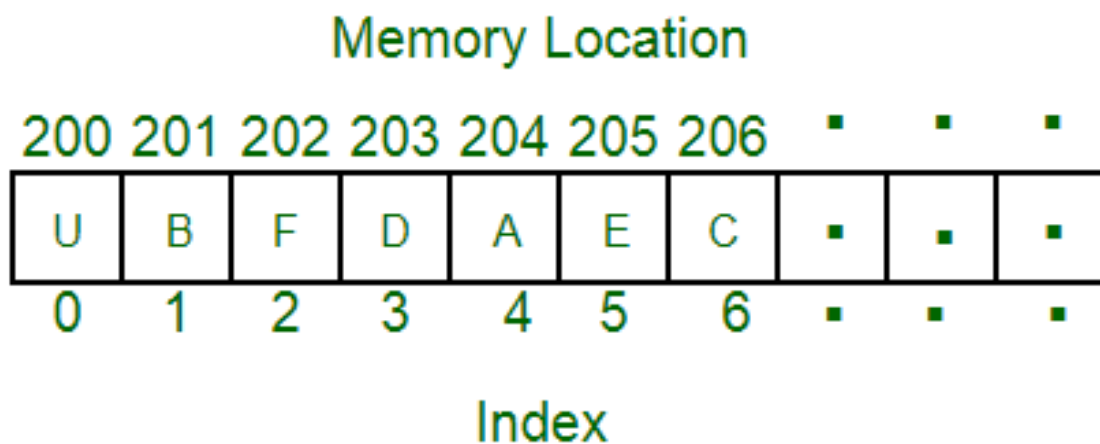
We are discuss some topics of data structure are given below :

- Array
- Stack
- Queue
- Linked List

1. Array

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value.

The memory location of the first element of the array



The above image can be looked as a top-level view of a staircase where you are at the base of the staircase. Each element can be uniquely identified by their index in the array.

There are two types of array:

1. 1D array
2. 2D array.

1.1 1D Array

An array is stored such that the position of each element can be computed from its index by a mathematical formula. The simplest type of data structure is a linear array, also called one-dimensional array.

```
Int main()  
for(i=0; i<n; i++){  
    Scanf("%d",&a[i]);  
}
```

1.2 2D Array

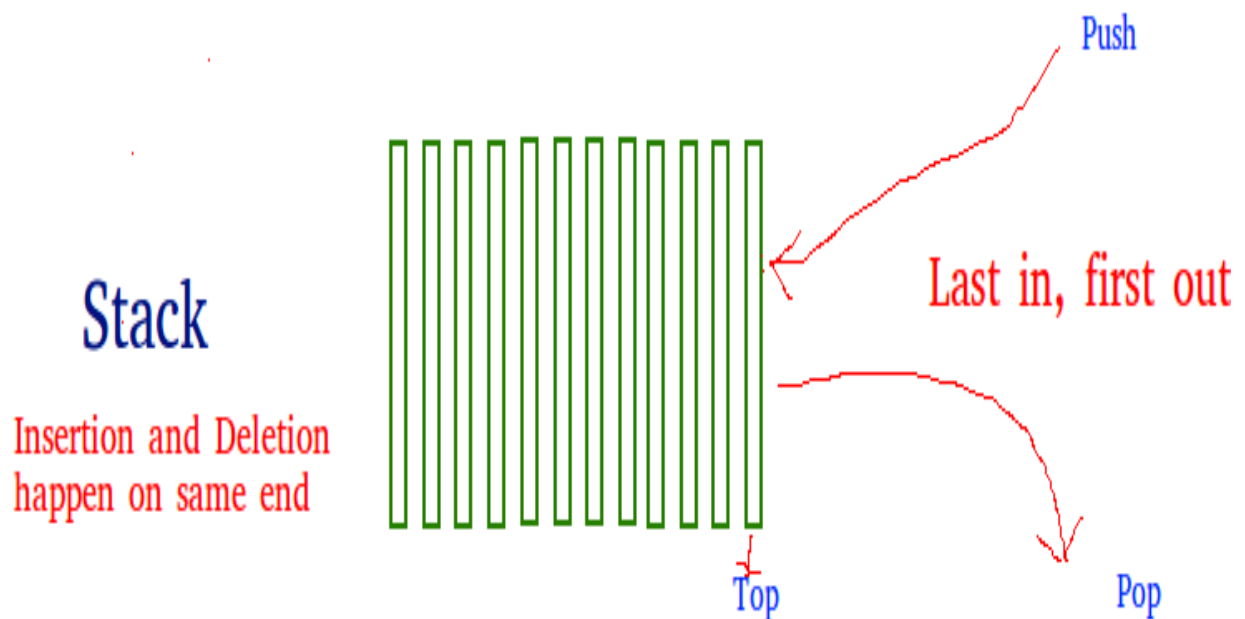
Two-dimensional (2D) arrays are indexed by two subscripts, one for the row and one for the column.

```
int a[10][10]
```

```
for(i=0; i<rows; i++){  
for(j=0; j<columns++;){  
}  
}
```

2. Stack

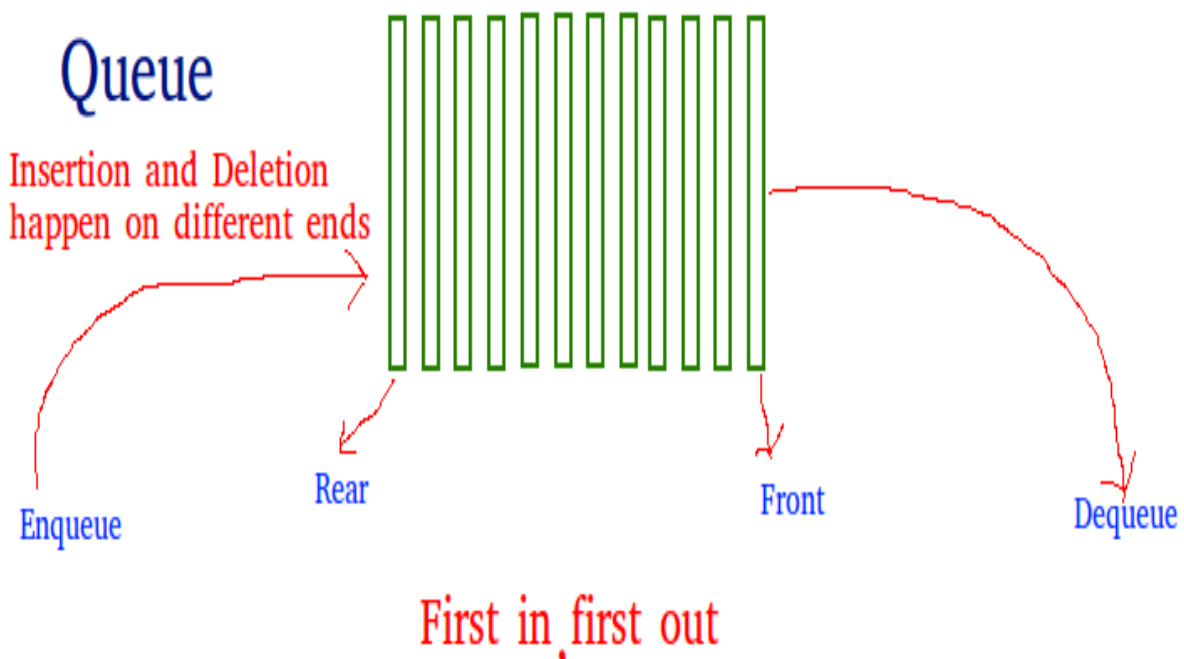
Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out).



There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed

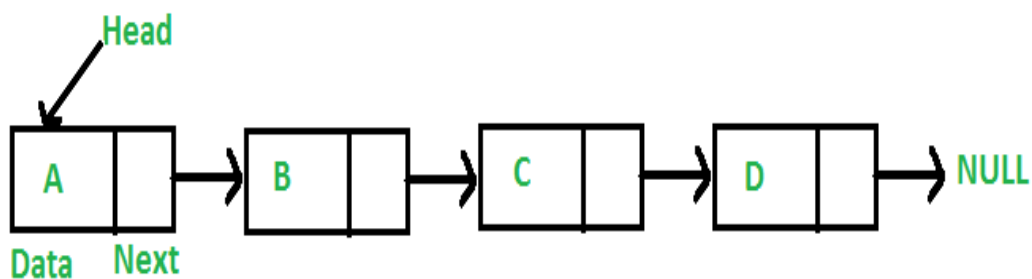
3. Queue

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between stack and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.



4. Linked List

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:



In simple words, a linked list consists of nodes where each node contains a data field and a reference to the next node in the list.

Linked list are two types:

1. Single linked list ,
2. Double linked list.

4.1 Single linked list

It is the most common. Each node has data and a pointer to the next node.



Node is represented as:

```
struct node {  
    int data;  
    struct node *next;  
}
```

4.2 Double linked list

We add a pointer to the previous node in a doubly linked list.



A node is represented as:

```
struct node {  
    int data;  
    struct node *next;  
    struct node *prev;  
}
```


References

➤ GeeksforGeeks

<https://www.geeksforgeeks.org/data-structures/>

➤ Programiz

<https://www.programiz.com/>