CSE1062 | CCS1063 ‘Practicals’ {

[Fundamentals of Computer Programming]

< Tutorial Session 11 - Data Structures >

}

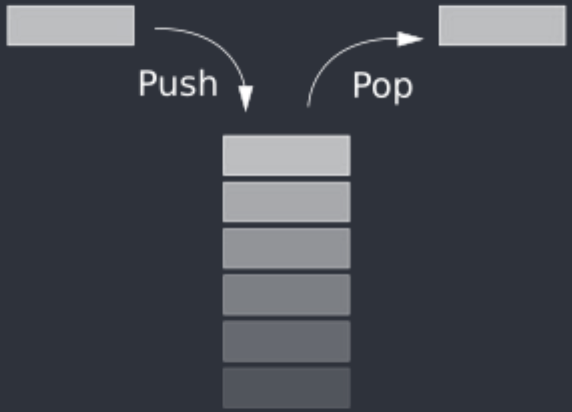
Stack

- It is an Abstract Data Type (ADT)
- Abstract Data Type is a data type (a set of values & a collection of operations on those values) that is accessed only through an interface
- Elements may be inserted or removed at one end called top of the stack
- First In Last Out (FILO) data structure

https://www.programiz.com/c-programming/c-structures-pointers

Comprises of two basic operations

* Insert (push) a new item
* Delete (pop) the item that was most recently inserted.



Basic Stack Processing Operations

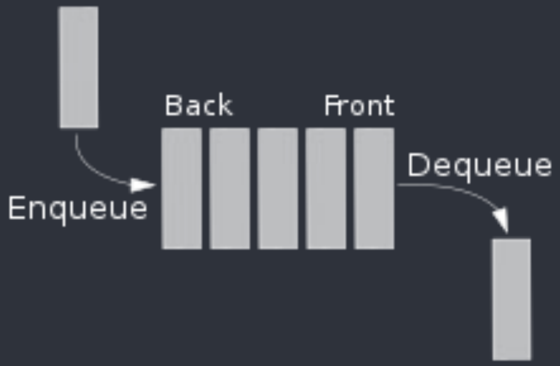
* Create a new node
* Stack initialization
* Push operation (Insertion)
* Pop operation (Deletion)
* Is Empty?

Queue

* It is an Abstract Data Type (ADT)
* First element added to the queue will be the first one to be removed
* First In First Out (FIFO) data structure

Comprises of two basic operations

* Insert (put) a new item
* Delete (get) the item that was least recently inserted



Basic Queue Processing Operations

* Create a new node
* Queue initialization
* Put operation (Insertion)
* Get operation (Deletion)
* Is Empty?

Linked List

* Data structure is composed of nodes
* Each node has
    ○ A data component
    ○ A link
* The data component could be any basic or
* structured data type
* Links are pointers to nodes

Linked List…

- A dynamic data structure whose elements linked one another through pointers
- Nodes are defined in terms of references to
- nodes

→Self-referent structures

- Advantage
  - Capability to rearrange the items efficiently

Linked List…

* Linked lists could be cyclic or doubly linked
* In most applications we work with One Dimensional List (Single Linked List)
* Single linked list
    ○ All the nodes except the last node each have exactly one link referring to them
* We use structures and pointers to represent nodes and links

Basic structure of a node

```
struct node // declare a structure called “node”
{
int data; // data
struct node *next; // declare node type pointer which can point
                      to another node
};
```

Memory Allocation

Memory allocation is a central consideration in the effective use of linked lists

Syntax:-

```
struct node *temp = (struct node*) malloc (sizeof(struct
node));
```

Basic linked list processing operations

* Create a new node
* List initialization
* Inserting nodes - Insert Front, Insert Rear and Insert Next
* Deleting nodes
* Searching
* Print list

```
Thanks; {

    ‘Do you have any questions?’

        < bgamage@sjp.ac.lk >

}
```