SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Computer Engineering Department
Program: MBA Tech. Sem III

Course: Data Structure
List of Experiments

w.e.f. 1/7/2019

LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

Experiment No.01

A.1 Aim:

Introduction to Data Structures. Select and implement appropriate data structure to solve the given problem.

A.2 Prerequisite:

1. Knowledge of different types of data structures.

2. Fundamental concepts of C\C++.

A.3 Outcome:

After successful completion of this experiment students will be able to

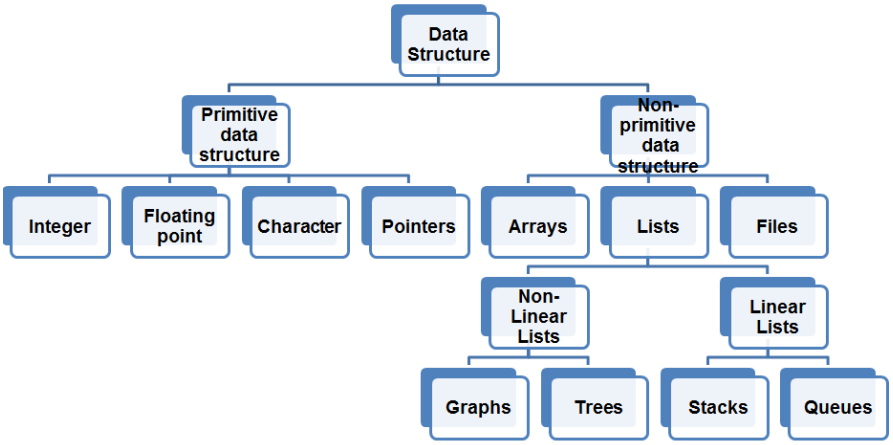
1. Identify the need of appropriate selection of data structure
2. Identify the steps of appropriate data structure selection.
3. Explore the effect of appropriate data structure selection.
4. Differentiate types of data structure based on their organization of data.
5. Implement appropriate selected data structure to solve the given problem
6. Enlist the applications of different data structure.

A.4 Theory:

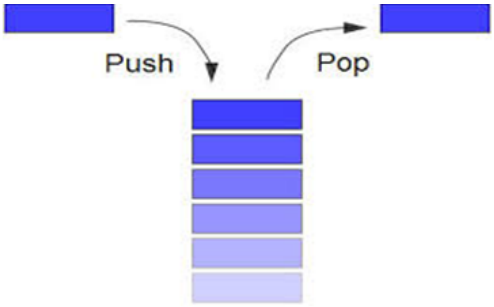
A.4.1. Introduction of Data structure

- The data may be organized into many different ways. The logical and mathematical model of a particular organization of data is called data structure.

- A data structure helps you to understand relationship of one data element with the other and organize it within the memory.
- Data structure specified following:
    - Organization of data
    - Accessing methods
    - Degree of associativity
    - Processing alternatives for information
- Classification of data structure:



- Primitive data structure:
    - Basic structures
    - Directly operated upon by the machine instructions
- Non- Primitive data structures:
    - Derived from primitive data structure.
    - Emphasize on structuring of a group of homogenous or heterogeneous data structure.
    - Ex: Arrays, Lists, Files
- Various data structure:



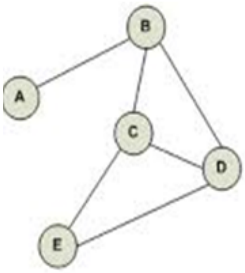


Fig 1. Undirected Graph

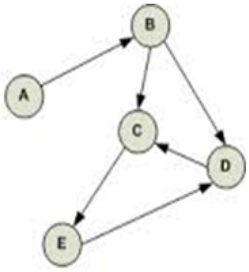
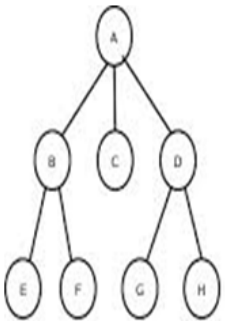
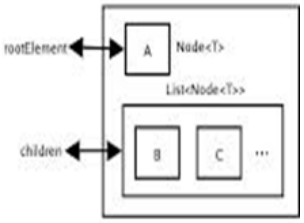


Fig 2. Directed Graph





A.5 Procedure/Algorithm:

A.5.1:

TASK 1:

Write a C/C++ program of array to perform following **(1D Array)**

i. Perform the reading and writing of the array.
ii. Find highest and lowest element in an array.
iii. Program to traverse an array.

TASK 2:

Identify suitable data structure for given scenarios. Specify reason for it.

1. To implement a system for reversing a word.
2. To implement printer spooler so that jobs can be printed in the order of the arrival.
3. To represent an image in a form of bitmap.
4. For representing a city region telephone network.
5. To store information about the directories and files in a system.
6. To implement back functionality in web browser.
7. To record sequence of all the pages browsed in one session.
8. To process network packets coming to the router.
9. To represent machines on internet and to find optimal path between source machine and destination machine to send data.

10.To represent connections\relations in social networking sites.
11.Google maps to travel from your home to office in minimum time.

****

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. N036	Name: Nischaya Sharma
Class : MBA(Tech) Computer	Batch : B2
Date of Experiment: 13/07/19	Date of Submission: 16/07/19
Grade :	Time of Submission: 1:30 AM
Date of Grading:	

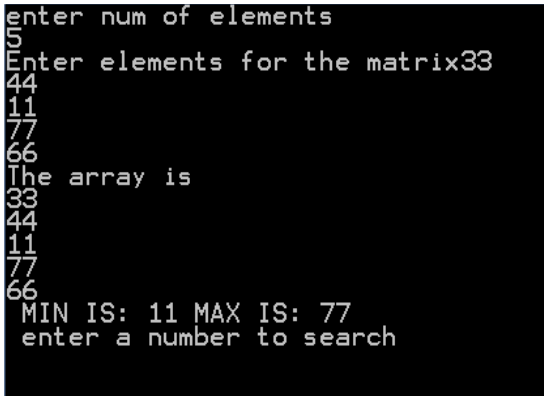
B.1 Software Code written by student: (Task 1)

```
#include<stdio.h>

int main()

{

        int a[100];

        int max1=0, min1=0, num=0, cc=0, nn=0;

        printf("enter num of elements\n");

        scanf("%d", &nn);

        printf("Enter elements for the matrix");

        for(int i=0;i<nn;i++){

                scanf("%d",&a[i]);
```

```
}
printf("The array is");
for(int i=0;i<nn;i++){
        printf("\n%d",a[i]);
}
max1= a[0];
min1= a[0];
for(int i=1; i<nn; i++){
        if(a[i]< min1){
                min1=a[i];
        }
}
printf("\n MIN IS: %d", min1);
for(int i=1; i<nn; i++){
        if(a[i]>max1){
                max1=a[i];
        }
}
printf(" MAX IS: %d", max1);
printf("\n enter a number to search");
scanf("%d", &num);
for(int i=0; i<nn; i++){
        if(a[i] == num){
                cc=1;
        }
```

```
        }
                if(cc==1){
                        printf("\nNUMBER IS PRESENT");
                }
                else{
                        printf("NOT PRESENT \n");
                }
        return 0;
}
```

B.2 Input and Output: **(Task 1)**



B.3 Identified data structure with reason: (Task 2)

1. To implement a system for reversing a word.
   ARRAYS - a single dimensional array is the most efficient way of reversing a word or accessing strings in any form.

2. To implement printer spooler so that jobs can be printed in the order of the arrival.

QUEUES- they follow the first IN first OUT procedure. Hence jobs can be printed as and when they arrive.

3. To represent an image in a form of bitmap.
   ARRAYS- a multi dimensional array can store the bits of the image that needs to be printed.
4. For representing a city region telephone network.
   TREES- to represent networking trees can extend their branches at various locations reaching out to a wider network.

5. To store information about the directories and files in a system.
   LINKED LISTS- the files and directories are connected in a system hence the linked lists stores them and keeps them all linked.

6. To implement back functionality in web browser.
   STACKS/ LINKED LISTS- they are vertical arrays following the last IN first OUT procedure, hence stacks will take us back to the last action or the last page that was open in the browser.

7. To record sequence of all the pages browsed in one session.
   ARRAYS- stores the pages of the browser in a single dimensional array.

8. To process network packets coming to the router.
   LINKED LISTS- the data sent to the router is in an order and is linked throughout.

9. To represent machines on internet and to find optimal path between source machine and destination machine to send data.
   GRAPHS- it connects different machines and finds the best possible way to connect them.

10. To represent connections\relations in social networking sites.
    LINKED LISTS/GRAPHS - it links the connections on the social media.

11. Google maps to travel from your home to office in minimum time.
    GRAPHS – they are basically used in finding the minimum distance between regions.

B.4 Observations and learning [w.r.t. all tasks]:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

B.5 Conclusion:

*(**Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

B.6 Question of Curiosity

Q1. Why appropriate selection of data structure is important in computer programming?

ANS. An appropriate selection of data structures is very important in programming as it makes the program easy to be handled, efficient and gives us a more robust method of fixing the bugs encountered in the program.
Different Data structures are used for different purposes and they need to be implemented accordingly so that the best use of them can be made.

****