FIRST SEMESTER MCA (2020 SCHEME) Practical Examination June 2021

20MCA135 DATA STRUCTURES LAB

Date : 30.06.2021 Time : 09.30 AM – 12.30 PM

Submitted by,
Afla Nazar
ICE20MCA-2003

Batch – A

Q1: Sorting of an integer array.

Algorithm

step 1: Initialize start
step 2: Initialize the variables i, j, a, n and an array with size 30.
step 3: Print "size of the array"
step 4: Read elements of the array as 'n'.
step 5: Print "elements of the array"
step 6: set i=0 Repeat step 7 and step 8 until i<n.
step 7: Read elements of ~~the~~ array
step 8: i = i+1
step 9: set i=0. Repeat step 10 to step 13 until i<n.
step 10: set j=i+1. Repeat step 11 until j<n.
step 11: if (number[i] > number[j]) then
a = number[i]
number[i] = number[j]
number[j] = a
step 12: j=j+1
step 13: i=i+1
step 14: Print "the sorted array"

step 15: set i=0. Repeat step 16 and step 17 until i<n.
step 16: Print number[i]
step 17: i=i+1
step 18: End.

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int i,j,a,n,number[30];
  clrscr();
  printf("Enter the size of the array: ");
  scanf("%d",&n);
  printf("Enter the elements of the array: ");
  for(i=0;i<n;i++)
      scanf("%d",&number[i]);
  for(i=0;i<n;i++)
  {
    for(j=i+1,j<n;j++)
    {
      if(number[i]>number[j])
      {
        a=number[i];
        number[i]=number[j];
        number[j]=a;
      }
    }
  }
  printf("The sorted array is:\n");
  for(i=0;i<n;i++)
      printf("%d\n",number[i]);
  getch();
}
```

Output

```
Enter the size of the array: 5
Enter the elements of the array: 4 1 6 3 2
The sorted array is:
1
2
3
4
6
```

Q2: Implement Prim's algorithm.

Algorithm

step 1: start
step 2: Initialize variables a, b, u, v, n, i, j, ne = 1.
step 3: Initialize visited[10] = {0}, min, mincost = 0,
cost[10][10]
step 4: Print "No. of nodes"
step 5: Read n.
step 6: Print "Adjacency matrix"
step 7: set i = 1. Repeat step 10 and step 11 until i <= n.
step 8: set j = 1 Repeat step 10 and step 12 until j <= n
step 9: Read cost
step 10: if (cost[i][j] == 0) then
cost[i][j] = 999.
step 11: i = i + 1 Repeat step
step 12: j = j + 1
step 13: set visited[1] = 1.
step 14: while (ne < n)
step 15: set i = 1, min = 999. Repeat step (a) upto step (d)
until i <= n.
step b: set j = 1 Repeat step (c) upto step (d) until j <= n.
step c: if (visited[i] != 0

min = cost[i][j]
a = u = i
b = v = j
step d: if (visited[u] == 0 || visited[v] == 0) then
print "edge and cost"
mincost = min
visited[b] = 1
~~step 15: Set cost[a][b] = cost[b][a] = 999.~~
~~step 16: Print "Minimum cost"~~
~~step 17: stop~~

step 15 : i = i + 1
step 16 : j = j + 1
step 17 : Set cost[a][b] = cost[b][a] = 999
step 18 : Print "Minimum cost"
step 19 : stop

Program

```
#include <stdio.h>
#include <conio.h>
void main()
{ int a,b,u,v,n,i,j,ne=1;
  int visited[10]={0}, min, mincost=0, cost[10][10];
  void main()
  {
    clrscr();
    printf("Enter the nodes:");
    scanf("%d",&n);
    printf("Enter the adjacency matrix:\n");
    for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
      scanf("%d",&cost[i][j]);
      if(cost[i][j]==0)
```

```
        cost[i][j] = 999;
    }
    visited[1] = 1;
    printf("\n");
    while (ne < n)
    {
        for (i=1, min=999; i<=n; i++)
            for (j=1; j<=n; j++)
                if (cost[i][j] < min)
                if (visited[i] != 0)
                {
                    min = cost[i][j];
                    a = u = i;
                    b = v = j;
                }
                if (visited[u] == 0 || visited[v] == 0)
                {
                    printf("\n edge %d: (%d %d) cost:
                        %d", ne++, a, b, min);
                    mincost += min;
                    visited[b] = 1;
                }
                cost[a][b] = cost[b][a] = 999;
        }
        printf("\n Minimum cost: %d", mincost);
        getch();
}
```

Output

```
Enter the no. of nodes: 6
Enter the adjacency matrix:
0 3 1 6 0 0
3 0 5 0 3 0
1 5 0 5 6 4
```

6 0 5 0 0 2
0 3 6 0 0 6
0 0 4 2 6 0

edge 1 : (13) cost : 1
edge 2 : (12) cost : 3
edge 3 : (25) cost : 3.
edge 4 : (36) cost : 4.
edge 5 : (64) cost : 2
Minimum cost : 13