First semester MCA (2020 scheme) practical
Examination June 2021
20MCA135 Data structures Lab

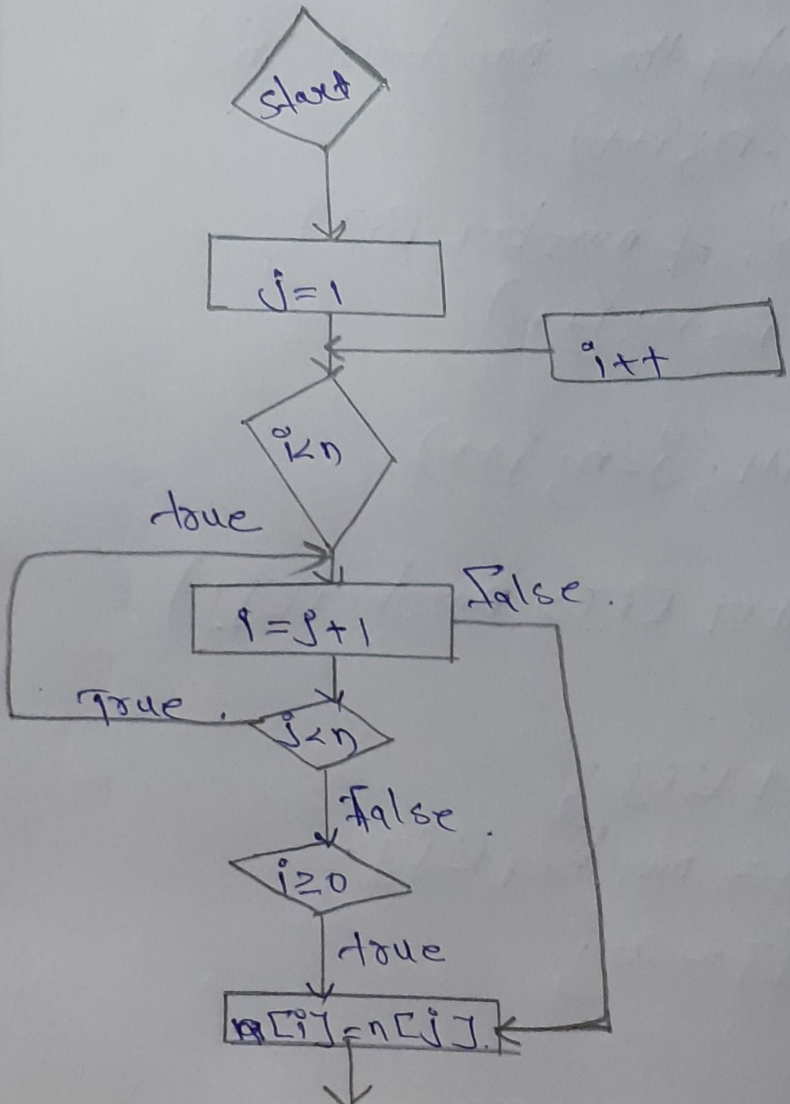
Date : 30.06.2021

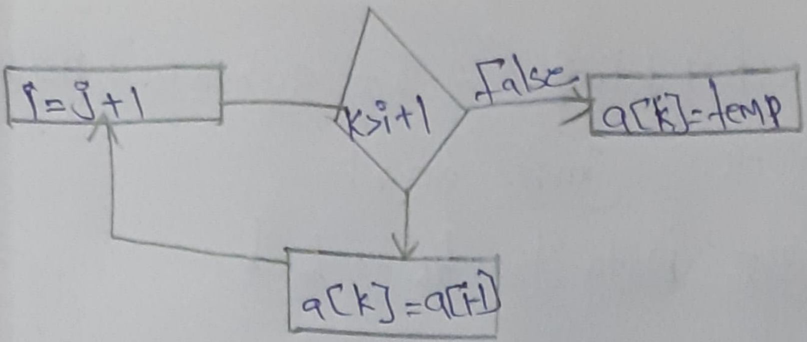
Time : 9.30AM - 12.30AM

Submitted by
Abija. KP
ICE20MCA-2001

1) Sorting of an Integer array ?

Flowchart





Program.

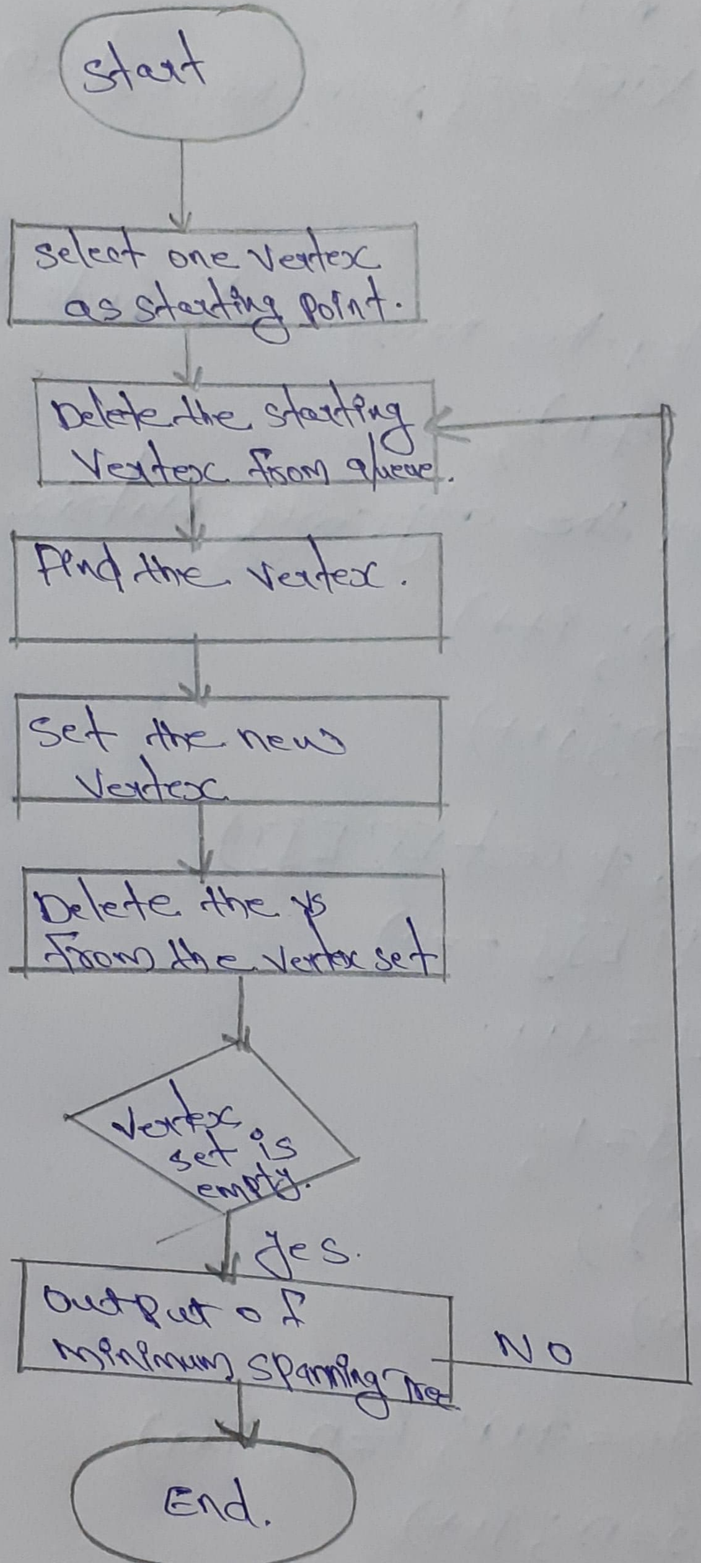
```
# include <stdio.h>
# include < conio.h>
   void Main ()
{ int i,j,a,n,number [30];
  clrscr ();
printf(" Enter the value of N \n");
 scanf ("%d", &n);
printf (" Enter the numbers \n");
 for(i=0; i<n; i++)
 scanf("%d", & number [i]);
  for(i=0; i<n; i++)
{ for (j= i+1; j<n; j++)
{ if (number [i]> number[j])
{
 a= number [i];
number [i]= number [j];
 number [j]= a;
}}}
printf (" The numbers arranged in ascending order\n")
for (i=0; i<n; i++)
printf ("%d \n", number [i]);
getch ();
}
```

O/P.

```
Enter the value of N: 8
Enter the numbers:
6
4
2
10
9
7
3
5
The numbers arranged in ascending order
2
3
4
5
6
7
9
10.
```

2) Implementing prims Algorithm.

Flowchart


start
select one vertex as starting point.
Delete the starting vertex from queue.
find the vertex.
set the new vertex
Delete the vs from the vertex set
vertex set is empty.
yes.
NO
output of minimum spanning tree
End.

Program

```
# include < stdio.h>
# include < conio.h>
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 number of 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);
min cost += min;
visited [b]=1;
}
cost [a][b] = cost [b][a]=999;
}
printf ("\n Minimum cost %d", min cost);
getch ();
}.
```

O/p.

```
Enter the number of nodes: 6.
Enter the adjacency Matrix:
0 3 1 6 0 0
3 0 5 0 3 0            Edge1: (1,3) cost:1
1 5 0 5 6 4            Edge :2: (1,2) cost :3
6 0 5 0 0 2            Edge:3(2,5) cost: 3
0 3 6 0 0 6            Edge 4(3,6) cost: 4
0 0 4 2 6 0.           Edge:5(6,4) cost:2
                    MiniMuM cost 13.
```