①

First Sem MCA (2020 SCHEME)
practical Examination June 2021
20MCA 135 DATA STRUCTURE LAB

Mahendran.K.S
ICE 20 MCA-2027
Date : 30 jun 2021
Time : 1:00 to 4:30

1, Sorting of integer array

2, Implement disjoint set operation

Answer

```
1, #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++)
      scanf("%d",&number[i]);
      for(i=0;i<n;i++)
       {
        for(j=i+1;j<n;j++)
          {
            if(numbers[i]>numbers[j])
              {
               a = numbers[i];
               numbers[i]=numbers[j];
               numbers[j]=a;
                }
           }
        }
        printf("The nubers arranged in ascending
                          Order are given below \n");
        for(i=0;i<n;++i)
        printf("%d\n",numbers[i]);
        getch();
       }
```

<u>Output</u>

```
Enter the value of N
5
Enter the numbers 1 8 5 2 4
The numbers arranged in ascending order
                                  are given below
 1
 2
 4
 5
 8
```

2 Disjoint set operation

```
#include <stdio.h>
#include <conio.h>
struct DisjSet
  {
   int parent[10];
   int rank[10];
   int n;
  }dis;
void makeSet()
 {
 int i;
 for (i=0; i<dis.n; i++)
  {
  dis.parent[i]=i;
  dis.rank[i]=0;
  }
 }
 void displaySet()
  {
   int i;
   printf("\n Parent Array\n");
   for(i=0; i<dis.n; i++)
    {
    printf("%d", dis.parent[i]);
    }
```

```
printf("\n Rank Array\n");
for (i=0; i<dis.n; i++)
  {
  printf("%d", dis.rank[i]);
  }
 printf("\n");
 }
int find (int x)
 {
if (dis.paret[x]!=x)
  {
  dis.parent [x]=find (dis.paret [x]);
  }
return dis.paret [x];
}
void Union (int x, int y)
 {
  int xset = find(x);
  int yset = find (y);
  If (xset == yset)
  return;
  if (dis.paret [xset] ~~[yset]~~ >dis.rank[yset])
   {
   dis.paret [x set] = yset;
   dis.rank[xset] =-1;
   }
```

```
else if(dis.rank[xset] <dis.rank[yset])
  {
  dis.parent[xset]=yset;
  dis.rank[xset]=-1;
  }
 else
 {
 dis.parent[yset]=xset;
 dis.rank[xset]=dis.rank[xset]+1;
 dis.rank[yset]=-1;
 }
 }
 int main()
   {
   int x,y,n,ch,wish;
   clrscr();
   printf("How many elements?");
   scanf("%d",&dis.n)
   makeSet();
   do
   {
   printf("\n ___ Menu___\n");
   printf("1.Union \n 2.Find \n 3.Display \n");
   printf("enter choice \n");
   scanf("%d",&ch);
   switch(ch)
   {
```

```
case 1:
printf("Enter elements to perform union:");
scanf("%d %d", &x, &y)
Union(x,y);
break;
case 2;
printf("Enter elements to check if connected components:")
scanf("%d %d", &x, &y);
if (find(x)= = find(y))
printf("connected components \n");
else
printf("Not connected components");
break;
case 3:
  displaySet ();
  break;
 }
printf("\n Do you wish to continue? (1/0) \n");
Scanf("%d", &wish);
}
while(wish==1);
return 0;
}
```

Output

```
How many elemets? 4
  MENU
1. Union
2. Find
3. Display
enter choice
 1
Enter elements to perform union 2 3

Do you wish to continue?.(1/0)
      MENU
1. Union
2. Find
3. Display
enter choice
2
Enter elements to check if connected compounts 3
Not connected compounts.
Do you wish to continue?.(1/0)
1
      MENU
1. Union
```

```
2. Find
3. Display
enter choice
3
parent Array
0 1 2 2
Rank Array
0 0 1 -1
Do you wish to continue?.(1,0)
```