**Slip 3**

Q.1) Write a program to sort a given set of elements using the Quick sort method and determine the time required to sort the elements

#include<stdio.h>

#include <time.h>

void quicksort(int x[20],int first,int last)

{

      int pivot,j,temp,i;

      if(first<last)

      {

            pivot=first;

            i=first;

            j=last;

            while(i<j)

            {

                  while(x[i]<=x[pivot]&&i<last)

                        i++;

                  while(x[j]>x[pivot])

                        j--;

                  if(i<j)

                  {

                        temp=x[i];

                        x[i]=x[j];

                        x[j]=temp;

                  }

            }

            temp=x[pivot];

            x[pivot]=x[j];

            x[j]=temp;

            quicksort(x,first,j-1);

            quicksort(x,j+1,last);

      }

}

int main()

{

      int x[20],size,i;

      printf("\tQuick sort\n");

      printf("-----------------------------------\n");

      printf(" How many numbers you want to sort?: ");

      scanf("%d",&size);

      printf("\n Enter %d elements: \n",size);

      for(i=0;i<size;i++)

            scanf("%d",&x[i]);

       clock\_t s,e,z;

        s=clock();

      quicksort(x,0,size-1);

      printf("\n Sorted elements after applying quick sort: \n\n");

      for(i=0;i<size;i++)

            printf(" %d",x[i]);

    e=clock();

    z=e-s;

    printf("\nTime taken: %f" , (float) z /CLOCKS\_PER\_SEC);

    printf(" seconds");

    return 0;

}

Q.2) Write a program to find Minimum Cost Spanning Tree of a given undirected graph using Prims algorithm

#include<iostream>

using namespace std;

int ne=1,min\_cost=0;

int main()

{

    int n,i,j,min,cost[20][20],a,u,b,v,source, visited[20];

    cout<<"Enter the no. of nodes:";

    cin>>n;

    cout<<"Enter the cost matrix:\n";

    for(i=1;i<=n;i++)

    {

        for(j=1;j<=n;j++)

        {

            cin>>cost[i][j];

        }

    }

    for(i=1;i<=n;i++)

    visited[i]=0;

    cout<<"Enter the root node:";

    cin>>source;

    visited[source]=1;

    cout<<"\nMinimum cost spanning tree is\n";

    while(ne<n)

    {

        min-999;

        for(i=1;i<=n;i++)

        {

            for(j=1;j<=n;j++)

            {

                if(cost[i][j]<min)

                    if(visited[i]=0)

                        continue;

                    else

                    {

                        min=cost[i][j];

                        a=u=i;

                        b=v=j;

                    }

                }

            }

            if(visited[u]==0||visited[v]==0)

            {

                cout<<"\nEdge"<<ne++<<"\t("<<a<<"->"<<b<<")="<<min;

                min\_cost-min\_cost+min;

                visited[b]=1;

            }

            cost[a][b]=cost[b][a]=999;

        }

    cout<<"\nMinimum cost="<<min\_cost<<"\n";

}