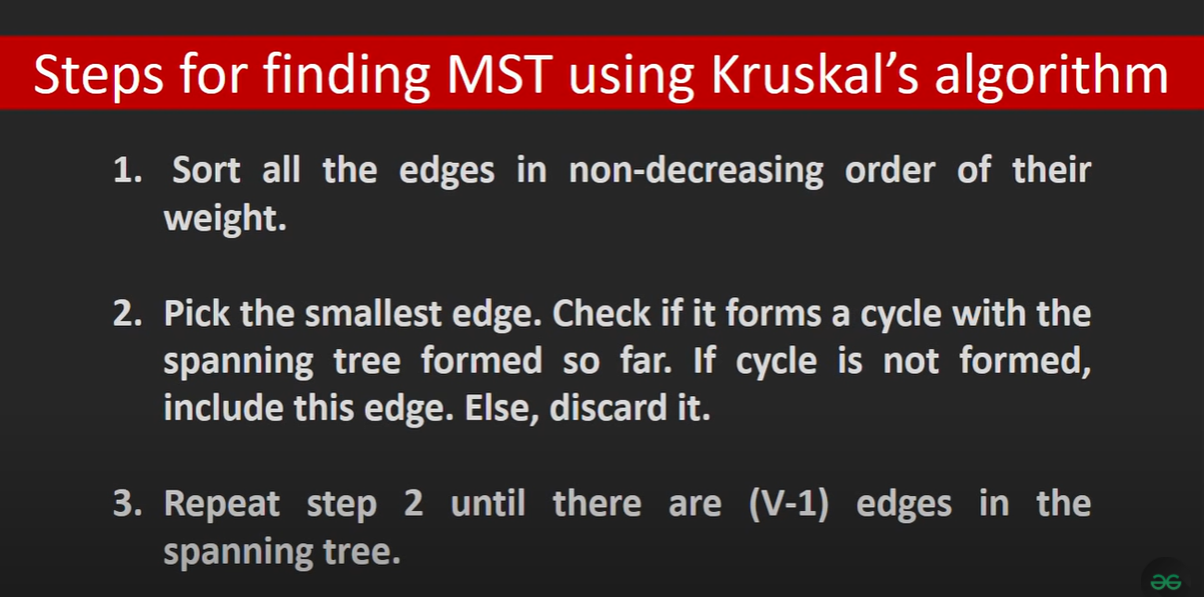
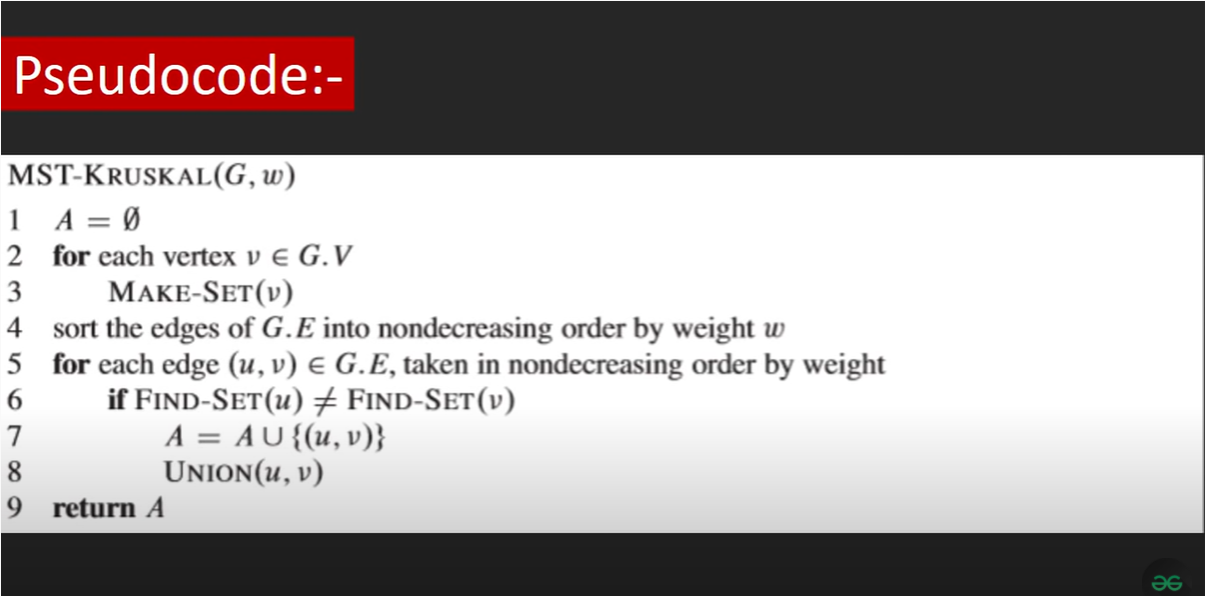
**PROGRAM 15: Kruskals algorithm**

Find Minimum Cost Spanning Tree of a given undirected graph using Kruskals algorithm.





**ALGORITHM:** kruskals(c[1…n,1…n])

//To compute the minimum spanning tree of a given weighted undirected graph using Kruskal’s

// algorithm

//Input: An nXn cost matrix c[1…n,1….n]

//Output: minimum cost of spanning tree of given undirected graph

ne🡨0

mincost🡨0

**for** i🡨1 to n **do**

     parent[i]🡨0

**end for**

**while** ne!=n-1 **do**

     min🡨9999

**for** i🡨1 to n **do**

**for** j🡨1 to n **do**

**if** c[i,j]<min

                   min🡨c[i,j]

                   u🡨i

                   a🡨i

                v🡨j

                b🡨j

**end if**

**end for**

**end for**

**while** parent[u]!=0 **do**

             u🡨parent[u]

**end while**

**while** parent[v]!=0 **do**

             v🡨parent[v]

**end while**

**if** u!= v

       write a,b,min

       parent[v]🡨u

       ne🡨ne+1

       mincost🡨mincost+min

**end if**

    c[a,b]🡨9999

    c[b,a]🡨9999

**end while**

write mincost

**return**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

void kruskals();

int c[10][10];

int n; // No of edges

int main()

{

int i,j;

printf("\nEnter the no. of vertices: ");

scanf("%d",&n);

printf("\nEnter the cost matrix:\n");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

scanf("%d",&c[i][j]);

}

}

kruskals();

}

void kruskals()

{

int i,j,u,v,a,b,min;

int ne=0,mincost=0;

int parent[10];

for(i=1;i<=n;i++)

{

parent[i]=0;

}

while(ne!=n-1)

{

min=9999;

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

if(c[i][j]<min)

{

min=c[i][j];

u=a=i;

v=b=j;

}

}

}

while(parent[u]!=0)

{

u=parent[u];

}

while(parent[v]!=0)

{

v=parent[v];

}

if(u!=v)

{

printf("\n\t%C <---> %C = %d\n",a+65,b+65,min);

parent[v]=u;

ne=ne+1;

mincost=mincost+min;

}

c[a][b]=c[b][a]=9999;

}

printf("\n Minimum cost is =%d",mincost);

}

                                        ==========Output=============

Enter the no. of vertices:   6

Enter the cost matrix:

9999          3    9999    9999          6           5

      3    9999          1    9999    9999           4

9999          1    9999          6    9999           4

9999          6          6    9999          8           5

      6    9999    9999          8    9999           2

      5          4          4          5          2     9999

B-----------> C = 1

E-----------> F = 2

A-----------> B = 3

B-----------> F = 4

F-----------> D = 5

Minimum cost = 15

