PRIMS :

#include<stdio.h>

#include<conio.h>

void prims ();

int c[10][10], n;

void

main ()

{

int i, j;

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

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]);

}

}

prims ();

}

void

prims ()

{

int i, j, u, v, min;

int ne = 0, mincost = 0;

int elec[10];

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

{

elec[i] = 0;

}

elec[1] = 1;

while (ne != n - 1)

{

min = 9999;

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

{

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

{

if (elec[i] == 1)

{

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

{

min = c[i][j];

u = i;

v = j;

}

}

}

}

if (elec[v] != 1)

{

printf ("\n%d----->%d=%d\n", u, v, min);

elec[v] = 1;

ne = ne + 1;

mincost = mincost + min;

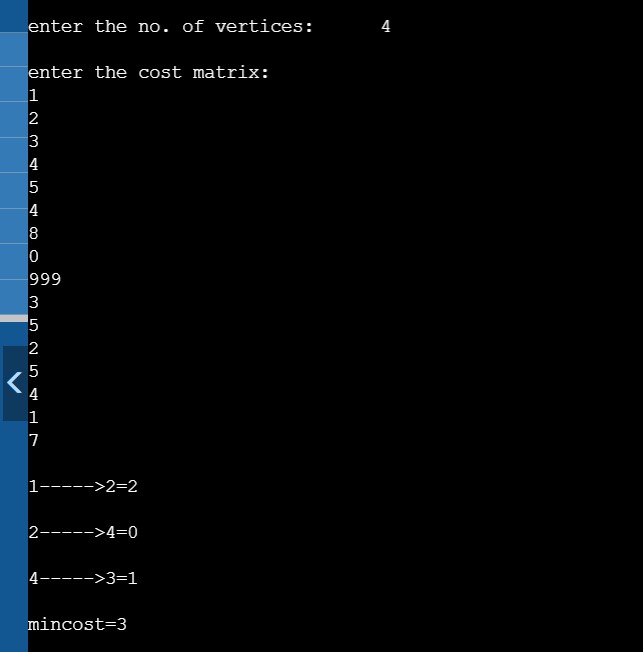
}

c[u][v] = c[v][u] = 9999;

}

printf ("\nmincost=%d", mincost);

}



KRUSKALS :

#include<stdio.h>

#include<conio.h>

void kruskals ();

int c[10][10], n;

void

main ()

{

int i, j ;

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

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%d----->%d=%d\n", a, b, min);

parent[v] = u;

ne = ne + 1;

mincost = mincost + min;

}

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

}

printf ("\nmincost=%d", mincost);

}

