**DATA STRUCTURES**

**DAY -5**

**DIJISTRA ALGORITHM**

#include<stdio.h>

#include<conio.h>

#define INF 9999

void diji(int a[10][10], int n, int stn) {

int dist[10], p[10], vst[10];

int mdist, cnt, nn, i, j;

for (i = 0; i < n; i++) {

dist[i] = a[stn][i];

p[i] = stn;

vst[i] = 0;

}

dist[stn] = 0;

vst[stn] = 1;

cnt = 0;

while (cnt < n - 1) {

mdist = INF;

for (i = 0; i < n; i++) {

if (!vst[i] && dist[i] < mdist) {

mdist = dist[i];

nn = i;

}

}

vst[nn] = 1;

for (i = 0; i < n; i++) {

if (!vst[i]) {

if (mdist + a[nn][i] < dist[i]) {

dist[i] = mdist + a[nn][i];

p[i] = nn;

}

}

}

cnt++;

}

for (i = 0; i < n; i++) {

if (i != stn) {

printf("\n Distance of %d = %d ", i, dist[i]);

printf("\n Path = %d ", i);

j = i;

do {

j = p[j];

printf(" <- %d ", j);

} while (j != stn);

}

}

}

int main() {

int a[10][10], i, j, n, u;

printf("Enter number of vertices: ");

scanf("%d", &n);

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

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

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

if (a[i][j] == 0)

a[i][j] = INF;

}

}

printf("Enter start node: ");

scanf("%d", &u);

diji(a, n, u);

return 0;

}

