Rajath. M. K
1BM18CS079

CN-Lab

Dijkstra's Algorithm to compute shortest path:

```
#include<iostream>
#include<conio>
#include<string>
#include<process>
#include<math.h>
# define N 6

int dijkstra(int cost[][N], int source, int target);
int dijkstra( int cost[][N], int source, int target);
{
    int dist[N], prev[N], selected[N]={0}, i, m, min;
    int start, dj;
    char path[N]

    for(int i=1; i<N; i++){
        dist[i] = IN;
        prev[i] = -1;
    }
start = source;
selected[start] = 1;
dist[start] = 0;
```

1BM18CS079

```
while (selected[target] == 0){
    min = INF;
    m = 0;
    for(i = 1; i<N; i++){
        d = dist[start] + cost[start][i];
        if (d < dist[i] && selected[i] == 0){
            dist[i] = d;
            prev[i] = start;
        }
        if (min > dist[i] && selected[i] == 0){
            min = dist[i];
            m = i;
        }
    start = m;
    selected[start] = 1
}
start = target;
j = 0;
while(start != -1){
    path[j++] = start + 65;
    start = prev[start];
}
path[j] = '\0'
strrev(path);
cout << path;
return dist[target];
}
```

1BM18CC079

```
int main(){
    int cost[N][N], i,j, w,ch,co;
    int source, target, x, y;
    cout<<"shortest path :";
    for(i=1; i<N; i++){
        for(j=1; j<N; j++){
            cost[i][j] = IN;
            for(x=1; x<N; x++){
                for(y=x+1; y<N; y++){
                    ~~printf("Enter weight~~
                    cout<<"Enter weight between node"
                    <<x<<" & "<<y;
                    cin>>w;
                    cost[x][y] = cost[y][x] = w;
                }
            }
        ~~printf("~~
        cout<<"Enter the source";
        cin>>source;
        cout<<"Enter target";
        cin>>target;
        co = dijkstra(cost, source, target);
        cout<<"shortest path"<<co;
    }
```