

CN Lab 10 - Dijkstra's

Rajath M.K

18M18CCE079

```
#include <stdio.h>
```

```
#define V 9
```

```
int minDist (int dist[], bool sptset[])  
{
```

```
    int min = INT_MAX, min_index;
```

```
    for (int v = 0; v < V; v++) {
```

```
        if (sptset[v] == false && dist[v] <= min) {
```

```
            min = dist[v], min_index = v;
```

```
        return min_index;
```

```
    }
```

```
void printSolution (int dist[]) {
```

```
    printf ("vertex\t\t distance from source \n")
```

```
    for (int i = 0; i < V; i++)
```

```
        printf ("%d\t\t %d", i, dist[i]);
```

```
}
```

```
void dijkstras (int graph[V][V], int src) {
```

```
    int dist[V];
```

```
    bool sptset[V];
```

```
    for (int i = 0; i < V; i++) {
```

```
        dist[i] = INT_MAX, sptset[i] = false;
```

dist[src] = 0

1BM18C5079

```
for (int count = 0; count < V-1; count++) {
```

```
    int u = minDistance(dist, sptSet);
```

```
    sptSet[u] = true;
```

```
    for (int v = 0; v < V; v++) {
```

```
        if (!sptSet[v] && graph[u][v] && dist[u]
```

```
            != INT_MAX
```

```
            && dist[u] + graph[u][v] < dist[v])
```

```
            dist[v] = dist[u] + graph[u][v];
```

```
    }
```

```
    printSolution(dist);
```

```
}
```

```
int main() {
```

```
    int graph[V][V];
```

```
    cout << "Enter graph:"
```

```
    for (i = 0; i < V; i++) {
```

```
        for (j = 0; j < V; j++) {
```

```
            cin >> graph[i][j];
```

```
        }
```

```
    if (kstraal(graph, 0))
```

```
        return 0;
```

```
}
```