

Dijkstra - Printing - Paths

```
#include <stdio.h>
#include <limits.h>
#define V 9

int minDistance(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 printPath(int parent[], int j){
    if (parent[j] == -1)
        return;

    printPath(parent, parent[j]);

    printf("%d ", j);
}

int printSolution(int dist[], int n, int parent[]){
    int src = 0;
    printf("Vertex\t Distance\t Path");
    for (int i = 1; i < V; i++){
        printf("\n%d -> %d \t\t %d\t\t %d ",
            src, i, dist[i], src);
        printPath(parent, i);
    }
}

void dijkstra(int graph[V][V], int src){
    int dist[V];
    bool sptSet[V];
    int parent[V];

    for (int i = 0; i < V; i++){
        parent[i] = -1;
    }
}
```

```

        dist[i] = INT_MAX;
        sptSet[i] = false;
    }

    dist[src] = 0;

    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] + graph[u][v] < dist[v]){
                parent[v] = u;
                dist[v] = dist[u] + graph[u][v];
            }
        }
    }

    printSolution(dist, V, parent);
}

int main(){
    int graph[V][V] = {{0, 4, 0, 0, 0, 0, 0, 8, 0},
                        {4, 0, 8, 0, 0, 0, 0, 11, 0},
                        {0, 8, 0, 7, 0, 4, 0, 0, 2},
                        {0, 0, 7, 0, 9, 14, 0, 0, 0},
                        {0, 0, 0, 9, 0, 10, 0, 0, 0},
                        {0, 0, 4, 0, 10, 0, 2, 0, 0},
                        {0, 0, 0, 14, 0, 2, 0, 1, 6},
                        {8, 11, 0, 0, 0, 0, 1, 0, 7},
                        {0, 0, 2, 0, 0, 0, 6, 7, 0}
    };

    dijkstra(graph, 0);
    return 0;
}

```