

## 24.Implementation of Shortest Path Algorithms using Dijkstra's Algorithm

```
#include <stdio.h>

#define INF 9999

#define MAX 10

void dijkstra(int graph[MAX][MAX], int n, int start) {

    int cost[MAX][MAX], dist[MAX], visited[MAX], count, minDist, nextNode, i, j;

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

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

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

                cost[i][j] = INF;

            else

                cost[i][j] = graph[i][j];

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

        dist[i] = cost[start][i];

        visited[i] = 0;

    }

    dist[start] = 0;

    visited[start] = 1;

    count = 1;

    while (count < n - 1) {

        minDist = INF;

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

            if (dist[i] < minDist && !visited[i]) {

                minDist = dist[i];

                nextNode = i;

            }

        visited[nextNode] = 1;

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

            if (!visited[i])

                if (minDist + cost[nextNode][i] < dist[i])

                    dist[i] = minDist + cost[nextNode][i];

    }
```

```

        count++;
    }

    printf("\nShortest distances from vertex %d:\n", start);

    for (i = 0; i < n; i++)
        if (i != start)
            printf("To vertex %d = %d\n", i, dist[i]);
}

int main() {

    int graph[MAX][MAX], n, i, j, start;

    printf("Enter number of vertices: ");

    scanf("%d", &n);

    printf("Enter the adjacency matrix (use 0 if no edge):\n");

    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);

    printf("Enter the starting vertex (0 to %d): ", n - 1);

    scanf("%d", &start);

    dijkstra(graph, n, start);

    return 0;
}

```

main.c	Output
<pre> 1 #include &lt;stdio.h&gt; 2 #define INF 9999 3 #define MAX 10 4 void dijkstra(int graph[MAX][MAX], int n, int start) { 5     int cost[MAX][MAX], dist[MAX], visited[MAX], count, minDist,         nextNode, i, j; 6     for (i = 0; i &lt; n; i++) 7         for (j = 0; j &lt; n; j++) 8             if (graph[i][j] == 0) 9                 cost[i][j] = INF; 10            else 11                cost[i][j] = graph[i][j]; 12    for (i = 0; i &lt; n; i++) { 13        dist[i] = cost[start][i]; 14        visited[i] = 0; 15    } 16    dist[start] = 0; 17    visited[start] = 1; 18    count = 1; 19 20    while (count &lt; n - 1) { 21        minDist = INF; 22        for (i = 0; i &lt; n; i++) 23            if (dist[i] &lt; minDist &amp;&amp; !visited[i]) { 24                minDist = dist[i]; 25                nextNode = i; 26            } 27 </pre>	<pre> Enter number of vertices: 5 Enter the adjacency matrix (use 0 if no edge): 1 2 3 6 5 4 5 8 9 7 5 2 3 6 4 2 4 5 3 5 4 4 5 2 3 6 Enter the starting vertex (0 to 4): 1  Shortest distances from vertex 1: To vertex 0 = 4 To vertex 2 = 7 To vertex 3 = 9 To vertex 4 = 7  === Code Execution Successful === </pre>