

NAME:	Harshal Chawan
UID:	2021300019
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	06
DATE OF PERFORMANCE	27/03/2023
DATE OF SUBMISSION	03/04/2023
AIM:	To find shortest path using Dijkstra's Algorithm.
PROBLEM STATEMENT 1:	shortest path using Dijkstra's Algorithm and prim's algorithm.
ALGORITHM and THEORY:	<pre> function Dijkstra(<i>Graph</i>, <i>source</i>): 2 3 for each vertex <i>v</i> in <i>Graph.Vertices</i>: 4 <i>dist</i>[<i>v</i>] ← INFINITY 5 <i>prev</i>[<i>v</i>] ← UNDEFINED 6 add <i>v</i> to <i>Q</i> 7 <i>dist</i>[<i>source</i>] ← 0 8 9 while <i>Q</i> is not empty: 10 <i>u</i> ← vertex in <i>Q</i> with min <i>dist</i>[<i>u</i>] 11 remove <i>u</i> from <i>Q</i> 12 13 for each neighbor <i>v</i> of <i>u</i> still in <i>Q</i>: 14 <i>alt</i> ← <i>dist</i>[<i>u</i>] + <i>Graph.Edges</i>(<i>u</i>, <i>v</i>) 15 if <i>alt</i> < <i>dist</i>[<i>v</i>]: 16 <i>dist</i>[<i>v</i>] ← <i>alt</i> </pre>

	<pre> 17 prev[v] ← u 18 19 return dist[], prev[] </pre>
PROGRAM:	<pre> #include <limits.h> #include <stdbool.h> #include <stdio.h> int V; 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 printSolution(int dist[]) { printf("Vertex \t\t Distance from Source\n"); for (int i = 0; i < V; i++) printf("%d \t\t\t %d\n", i, dist[i]); } void dijkstra(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; </pre>

```

for (int count = 0; count < V - 1; count++) {
    int u = minDistance(dist, sptSet);

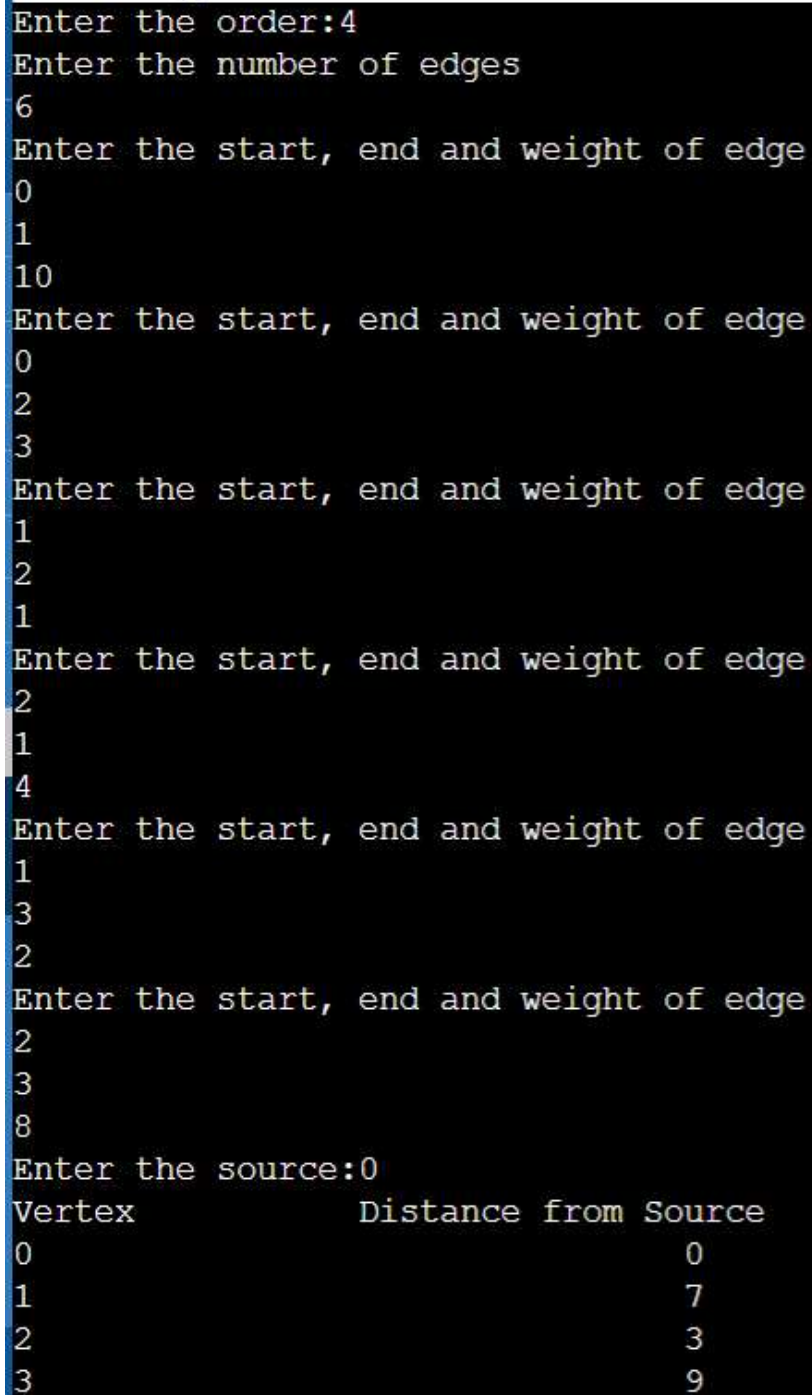
    sptSet[u] = true;

    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()
{
    printf("Enter the order:");
    scanf("%d",&V);
    int graph[V][V];
    for(int i=0;i<V;i++)
    {
        printf("Elements of row number %d:",(i+1));
        for(int j=0;j<V;j++)
        {
            scanf("%d",&graph[i][j]);
        }
    }
    dijkstra(graph, 0);
    return 0;
}

```

OUTPUT:

```
Enter the order:4
Enter the number of edges
6
Enter the start, end and weight of edge
0
1
10
Enter the start, end and weight of edge
0
2
3
Enter the start, end and weight of edge
1
2
1
Enter the start, end and weight of edge
2
1
4
Enter the start, end and weight of edge
1
3
2
Enter the start, end and weight of edge
2
3
8
Enter the source:0
Vertex          Distance from Source
0                0
1                7
2                3
3                9
```

CONCLUSION:	By performing the above experiment i have successfully found the shortest part of different vertices from a single source using Dijkstra's algorithm.