# Exercise 10 Dijkstra's Algorithm

**10. Write a C++ program for implementation of Dijkstra's algorithm.**

**Objective:**

The objectives of this exercise enable you to find the shortest path between two vertices in a graph with the help of Dijkstra's algorithm.

**Procedure and description:**

This algorithm is used to find the shortest path between the two vertices in a weighted directed graph and it is also very popular and efficient to find each and every path from starting (source) to terminal vertices.

**Algorithm:**

$\mathbf{v_s}$ :Source vertex

$\mathbf{v_t}$. : Terminal vertex (end vertex)

$\mathbf{V_r}$ :vertex with label r

$\mathbf{v_{s,z}}$: vertex associated with s and z; z intermediate vertices to reach the Terminal vertex

$\mathbf{w(v_i, v_j)}$ : the weight associated with every edge $(v_i, v_j)$

Dijkstra's algorithm is given below in stepwise manner:

Step 1: Assign a temporary label 1 $(v_i) = \infty$ to all vertices except $v_s$

Step 2: Mark $v_s$. as permanent by assigning 0 label to it

$1(v_{s.z)} = 0$

Step 3: Assign value of $v_s$ to $v_r$ where $v_r$ is last vertex to be made permanent. $V_r = V_s$

Step 4: If $(v_i) > (v_k) + w(v_k, v_i)$.

$(v_i) = (v_k) + w(v_k, v_i)$.

Step 5: $v_r = v_i$

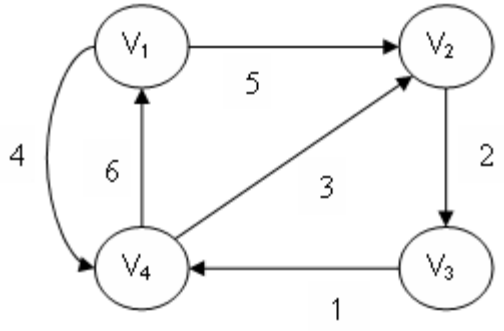
Step 6: If $v_t$ has temporary label, repeat step 4 to step 5 otherwise the value of $v_t$ is permanent label and is equal to the shortest path $v_s$ to $v_t$.

Step 7: Exit.

**Expected Output:**

**Input:**

After executing program. Enter input graph, vertices and distance between vertices as a two dimensional matrix. For better understanding a graphical representation is shown below




Enter any two vertex's between which the shortest path is to be found. Consider vertices as V1 to $V_3$ respectively.

**Output:** the shortest path between $V_1$ to $V_{3\ is}$: 7