

**Artificial Intelligence and Data Science Department.**

AOA / Even Sem 2021-22 / Experiment 5.

YASH SARANG.

47 / D6AD.

EXPERIMENT - 5.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Aim:** Write a program in C to implement Dijkstra's Single source

shortest path algorithm.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Theory:**

The Dijkstra algorithm is also called the single source shortest path algorithm. It is based on greedy technique.

The algorithm maintains a list visited[ ] of vertices, whose shortest distance from the source is already known.

If visited[1], equals 1, then the shortest distance of vertex i is already known. Initially, visited[i] is marked as, for source vertex.

At each step, we mark visited[v] as

1. Vertex v is a vertex at the shortest distance from the source vertex.

At each step of the algorithm, the shortest distance of each vertex is stored in an array distance[ ].

**Time Complexity:**

The program contains two nested loops each of which has a complexity of O(n). n is the number of vertices. So the complexity of the algorithm is O(n2).

**Auxiliary Space:** O(1)

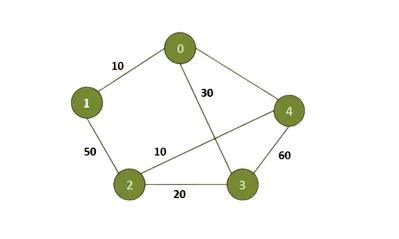
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CODE:**

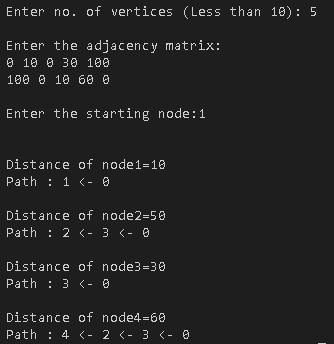
Code in the Djikstra.c file attached along with this doc.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**INPUT:**

****

**OUTPUT:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CONCLUSION:**

By performing this experiment, we conclude that :

The Time Complexity of the implementation is O(n^2), If the input graph is represented using an adjacency list, but it can be reduced to O(E log V) with the help of a binary heap.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**