

This code implements Dijkstra's Algorithm to find the shortest path between two nodes in a given weighted graph.

How it works:

1. Start at a chosen node, initially setting the distance to itself as zero and to all other nodes as infinite.
2. Select the unvisited node with the smallest known distance and explore its neighboring nodes, examining the connections from the current node.
3. Updates distances as it progresses, calculating new total distances to reach each neighbor.
4. If a shorter path to a neighbor is found, update the distance for that neighbor.
5. Repeat steps 2-4 until all reachable nodes have been processed.
6. After the main algorithm completes, the shortest path from the start node to any destination can be reconstructed using the recorded information.

Currently, the implementation of this feature contains a logic bug that causes the calculation of this algorithm to deviate from its intended functionality.

Algorithm expected outcomes:

For the graph provided in the code, the shortest path from node A to node E is [A, C, D, E], its distance is 5.

Algorithm actual outcomes:

For the graph provided in the code, the shortest path from node A to node E is [A, C, D, E], its distance is 2.