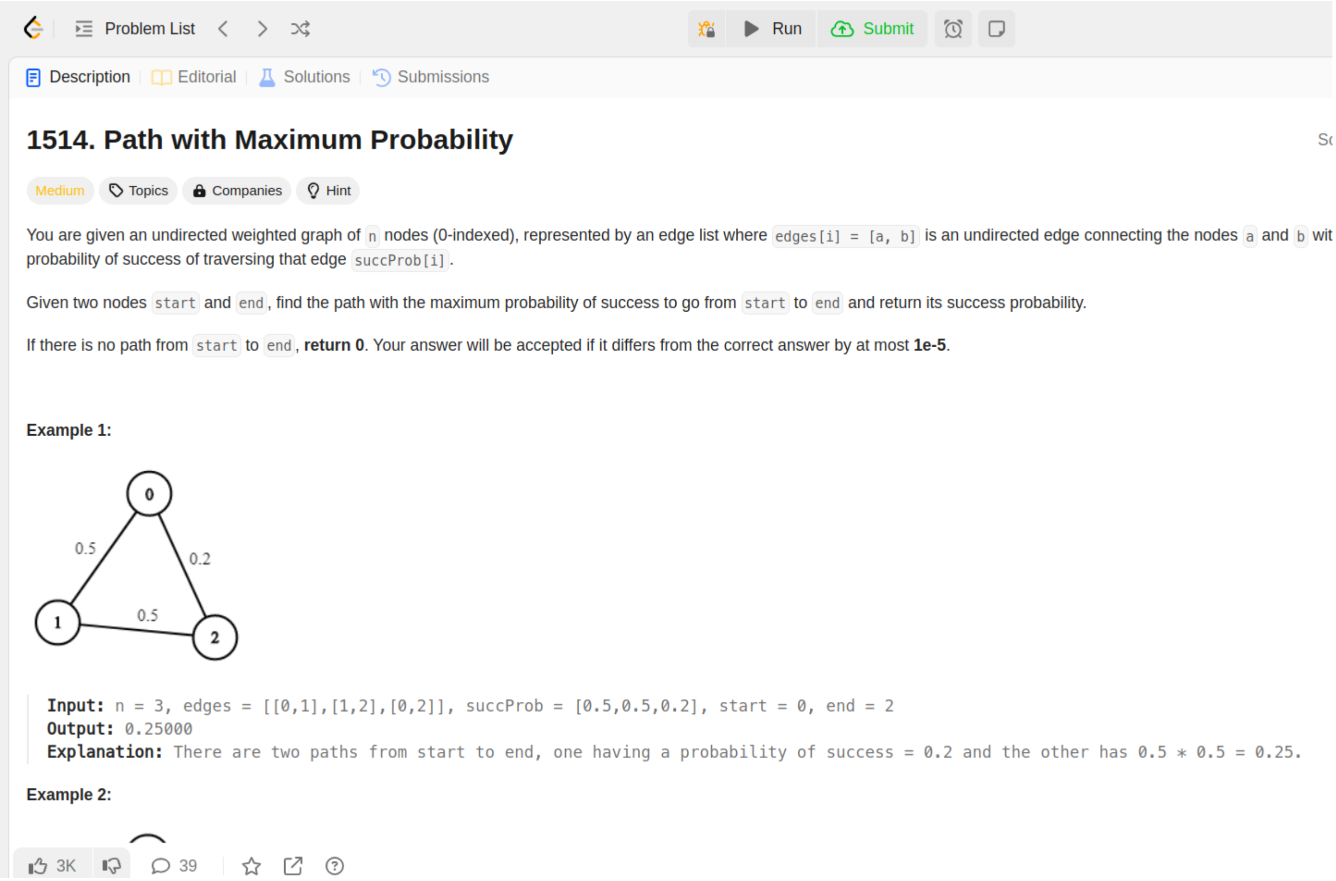
MCS – 253P ADVANCED PROGRAMMING AND PROBLEM SOLVING

LAB 8 Writeup(Path with Maximum Probability)

Aswin Sampath
saswin@uci.edu
53844684

Question)



1514. Path with Maximum Probability

You are given an undirected weighted graph of `n` nodes (0-indexed), represented by an edge list where `edges[i] = [a, b]` is an undirected edge connecting the nodes `a` and `b` wit probability of success of traversing that edge `succProb[i]`.

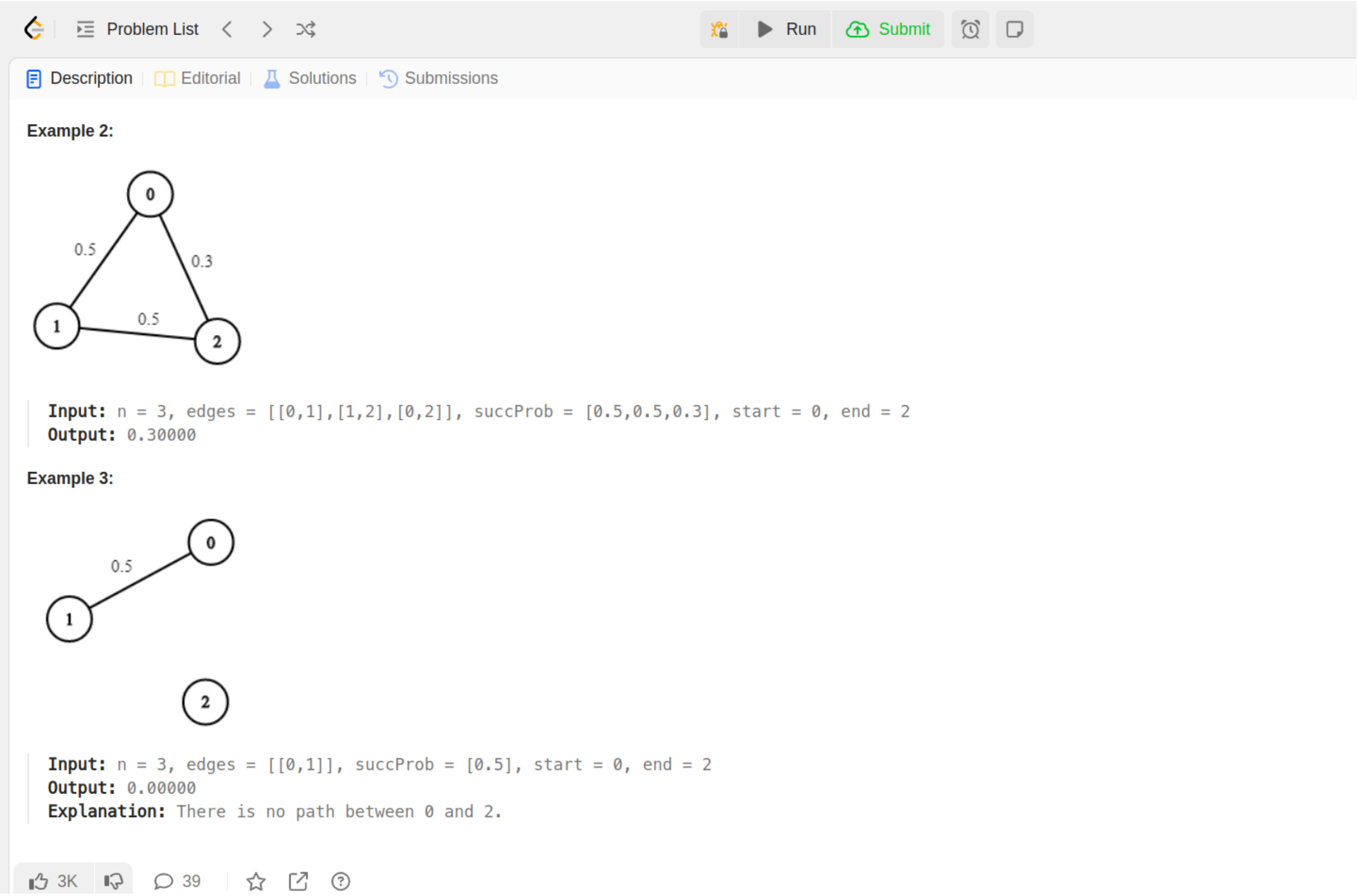
Given two nodes `start` and `end`, find the path with the maximum probability of success to go from `start` to `end` and return its success probability.

If there is no path from `start` to `end`, **return 0**. Your answer will be accepted if it differs from the correct answer by at most **1e-5**.

Example 1:

```
Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.2], start = 0, end = 2
Output: 0.25000
Explanation: There are two paths from start to end, one having a probability of success = 0.2 and the other has 0.5 * 0.5 = 0.25.
```

Example 2:



Example 2:

```
Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.3], start = 0, end = 2
Output: 0.30000
```

Example 3:

```
Input: n = 3, edges = [[0,1]], succProb = [0.5], start = 0, end = 2
Output: 0.00000
Explanation: There is no path between 0 and 2.
```

Understanding the Problem

The problem involves finding the path with the maximum probability of success to traverse an undirected weighted graph from a given start node to an end node. The graph is represented by an edge list, and each edge has an associated success probability.

Identifying Edge Cases

- An empty edge list: When the list of edges is empty, there are no connections, and the probability of success would be 0.
-
- A single-node graph: If there's only one node in the graph, it's not possible to traverse any edges, and the probability of success would be 0.

Effective Test Cases

A simple undirected graph with few edges and associated probabilities:

Nodes: 4

Edges: [[0,1],[1,2],[0,2],[2,3]]

Probabilities: [0.3,0.4,0.5,0.6]

Start Node: 0

End Node: 3

Expected Output: 0.3 (Assuming the path [0 -> 2 -> 3] has the maximum probability of success.)

A graph with disconnected components:

Nodes: 5

Edges: [[0,1],[1,2],[2,0],[3,4]]

Probabilities: [0.3,0.4,0.5,0.6]

Start Node: 0

End Node: 4

Expected Output: 0 (No path exists between the start and end nodes.)

Algorithmic Solution

- The provided C++ code implements Dijkstra's algorithm to find the maximum probability path in an undirected weighted graph:
- It initializes a priority queue to store nodes based on their probabilities and explores nodes in a greedy manner to calculate the maximum probability path.
- The code traverses the graph, updating the probabilities in the dist array whenever it finds a path with a higher probability.

Time and Space Complexity Analysis

The time complexity for this algorithm is O(E log V), where E represents the number of edges and V represents the number of vertices in the graph. It performs a priority queue-based traversal, considering each edge at most once.

The space complexity is O(V + E) for storing the graph and priority queue, where V is the number of vertices and E is the number of edges in the graph.