Problem 1129: Shortest Path with Alternating Colors

Problem Information

Difficulty: Medium
Acceptance Rate: 47.49%
Paid Only: No
Tags: Breadth-First Search, Graph

Problem Description

You are given an integer `n`, the number of nodes in a directed graph where the nodes are labeled from `0` to `n - 1`. Each edge is red or blue in this graph, and there could be self-edges and parallel edges.

You are given two arrays `redEdges` and `blueEdges` where:

* `redEdges[i] = [ai, bi]` indicates that there is a directed red edge from node `ai` to node `bi` in the graph, and * `blueEdges[j] = [uj, vj]` indicates that there is a directed blue edge from node `uj` to node `vj` in the graph.

Return an array `answer` of length `n`, where each `answer[x]` is the length of the shortest path from node `0` to node `x` such that the edge colors alternate along the path, or `-1` if such a path does not exist.

Example 1:

Input: n = 3, redEdges = [[0,1],[1,2]], blueEdges = [] **Output:** [0,1,-1]

Example 2:

Input: n = 3, redEdges = [[0,1]], blueEdges = [[2,1]] **Output:** [0,1,-1]

Constraints:

* `1 <= n <= 100` * `0 <= redEdges.length, blueEdges.length <= 400` * `redEdges[i].length == blueEdges[j].length == 2` * `0 <= ai, bi, uj, vj < n`

Code Snippets

C++:

```
class Solution {
public:
vector<int> shortestAlternatingPaths(int n, vector<vector<int>>& redEdges,
vector<vector<int>>& blueEdges) {

}
};
```

Java:

```
class Solution {
public int[] shortestAlternatingPaths(int n, int[][] redEdges, int[][]
blueEdges) {

}
}
```

Python3:

```
class Solution:
def shortestAlternatingPaths(self, n: int, redEdges: List[List[int]],
blueEdges: List[List[int]]) -> List[int]:
```