

難度 1.7 / 5

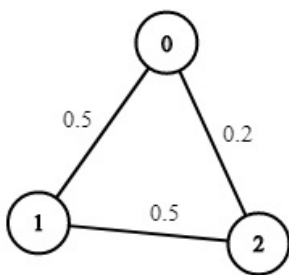
## 題目敘述

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

Given two nodes  $\text{start}$  and  $\text{end}$ , find the path with the maximum probability of success to go from  $\text{start}$  to  $\text{end}$  and return its success probability.

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

Example 1:

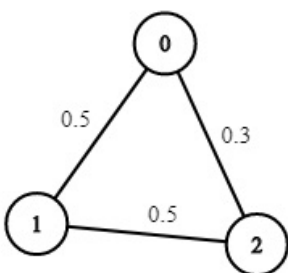


**Input:**  $n = 3$ ,  $\text{edges} = [[0,1],[1,2],[0,2]]$ ,  $\text{succProb} = [0.5,0.5,0.2]$ ,  $\text{start} = 0$ ,  $\text{end} = 2$

**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:



**Input:**  $n = 3$ ,  $\text{edges} = [[0,1],[1,2],[0,2]]$ ,  $\text{succProb} = [0.5,0.5,0.3]$ ,  $\text{start} = 0$ ,  $\text{end} = 2$

**Output:** 0.30000

## 參考答案

```
class Solution {
public:
    using pdi = pair<double, int>;
    using pid = pair<int, double>;
    vector<vector<pdi>> gp;
    double solve(int n, vector<vector<int>>& edges, vector<double>& succProb, int start_node,
int end_node) {
        gp.resize(n);
        const int m = (int)edges.size();
```

```

    for (int i = 0; i < m; ++i) {
        int a = edges[i][0], b = edges[i][1];
        double w = succProb[i];
        gp[a].emplace_back(b, w);
        gp[b].emplace_back(a, w);
    }

    vector<double> dis(n, 0.0);
    bool vis[n]; memset(vis, 0, sizeof(vis));
    priority_queue<pdi> pq;
    dis[start_node] = 1.0;
    pq.push({1.0, start_node});

    while (!pq.empty()) {
        auto [d, now] = pq.top(); pq.pop();
        if (vis[now]) continue;
        vis[now] = true;
        for (auto [i, w]: gp[now]) if (dis[i] < w * d) {
            dis[i] = w * d;
            pq.push({dis[i], i});
        }
    }

    return dis[end_node];
}
};

```