Description | Editorial | Solutions (1.1K) | Submissions

1514. Path with Maximum Probability

Hint

Medium | 2.5K | 52

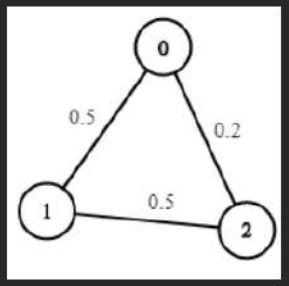
Companies

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` with a 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:

Java | Auto

```java
class Solution {
    public double maxProbability(int n, int[][] edges, double[] succProb, int start, int end) {
        // Adjacency list
        List<List<Pair<Integer, Double>>> adj = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            adj.add(new ArrayList<>());
        }

        for (int i = 0; i < edges.length; i++) {
            int u = edges[i][0];
            int v = edges[i][1];
            double p = succProb[i];
            adj.get(u).add(new Pair<>(v, p));
            adj.get(v).add(new Pair<>(u, p));
        }

```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

n =
3

edges =
[[0,1],[1,2],[0,2]]

succProb =
[0.5,0.5,0.2]

start =
0

Console | Run | Submit

Description | Editorial | Solutions (638) | Submissions

2652. Sum Multiples

Hint

Easy | 272 | 17

Companies

Given a positive integer `n`, find the sum of all integers in the range `[1, n]` **inclusive** that are divisible by `3`, `5`, or `7`.

Return *an integer denoting the sum of all numbers in the given range satisfying the constraint.*

Example 1:

```
Input: n = 7
Output: 21
Explanation: Numbers in the range [1, 7] that are divisible by 3, 5, or 7 are 3, 5, 6, 7. The sum of these numbers is 21.
```

Example 2:

```
Input: n = 10
Output: 40
Explanation: Numbers in the range [1, 10] that are divisible by 3, 5, or 7 are 3, 5, 6, 7, 9, 10. The sum of these numbers is 40.
```

Example 3:

```
Input: n = 9
Output: 30
Explanation: Numbers in the range [1, 9] that are divisible by 3, 5, or 7 are 3, 5, 6, 7, 9. The sum of these numbers is 30.
```

Constraints:

Java | Auto

```java
class Solution {
    public int sumOfMultiples(int n) {
        int count=0;

        for(int i=1; i<=n; i++){
            if(i%7==0){
                count+=i;
            }
            else if(i%5==0){
                count+=i;
            }
            else if(i%3==0){
                count+=i;
            }

        }
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

n =
7

Output

21

Expected

21

Contribute a testcase

Console | Run | Submit