75 Days of Code
Day 46
Problem no : 399
Problem Title :Evaluate Division
Type : Graph / DFS

You are given an array of variable pairs equations and an array of real numbers values, where equations[i] = [Ai, Bi] and values[i] represent the equation Ai / Bi = values[i]. Each Ai or Bi is a string that represents a single variable.

You are also given some queries, where queries[j] = [Cj, Dj] represents the jth query where you must find the answer for Cj / Dj = ?.

Return *the answers to all queries*. If a single answer cannot be determined, return -1.0.

Note: The input is always valid. You may assume that evaluating the queries will not result in division by zero and that there is no contradiction.

Note: The variables that do not occur in the list of equations are undefined, so the answer cannot be determined for them.

Example 1:

Input: equations = [["a","b"],["b","c"]], values = [2.0,3.0], queries = [["a","c"],["b","a"],["a","e"],["a","a"],["x","x"]]
Output: [6.00000,0.50000,-1.00000,1.00000,-1.00000]
Explanation:

Given: *a / b = 2.0*, *b / c = 3.0*
queries are: *a / c = ?*, *b / a = ?*, *a / e = ?*, *a / a = ?*, *x / x = ?*
return: [6.0, 0.5, -1.0, 1.0, -1.0]
note: x is undefined => -1.0
Example 2:

Input: equations = [["a","b"],["b","c"],["bc","cd"]], values = [1.5,2.5,5.0], queries = [["a","c"],["c","b"],["bc","cd"],["cd","bc"]]
Output: [3.75000,0.40000,5.00000,0.20000]

Example 3:

Input: equations = [["a","b"]], values = [0.5], queries = [["a","b"],["b","a"],["a","c"],["x","y"]]
Output: [0.50000,2.00000,-1.00000,-1.00000]

```
    }

    visited.delete(node);
    return -1.0;
}

const results: number[] = [];

for (const query of queries) {
    const [start, end] = query;
    const visited = new Set<string>();
    const result = dfs(start, end, visited);
    results.push(result);
}

return results;
};
```

```typescript
function calcEquation(equations: string[][], values: number[], queries: string[][]): number[] {
    const graph: Record<string, Record<string, number>> = {};

for (let i = 0; i < equations.length; i++) {
  const [numerator, denominator] = equations[i];
  const value = values[i];

  if (!(numerator in graph)) {
    graph[numerator] = {};
}
  if (!(denominator in graph)) {
    graph[denominator] = {};
  }

  graph[numerator][denominator] = value;
  graph[denominator][numerator] = 1 / value;
}

function dfs(node: string, target: string, visited: Set<string>): number {
  if (!(node in graph)) {
    return -1.0;
  }

  if (node === target) {
    return 1.0;
  }

  visited.add(node);

  for (const neighbor in graph[node]) {
    if (!visited.has(neighbor)) {
      const result = dfs(neighbor, target, visited);
      if (result !== -1.0) {
        return result * graph[node][neighbor];
      }
    }
  }
```

Accepted

Editorial + Solution

Runtime	Details	Memory	Details
58 ms		43.07 MB	
Beats 51.70% of users with TypeScript		Beats 84.09% of users with TypeScript	

More challenges

1 2 3 4 5 6 7 8 9 10 11 12

Shubham Agrahari

Shubham Agrahari