# 399. Evaluate Division

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.

## SOLUTION IN JAVA

class Solution {

public double[] calcEquation(List<List<String>> equations, double[] values,

List<List<String>> queries) {

double[] ans = new double[queries.size()];

Map<String, Map<String, Double>> graph = new HashMap<>();

for (int i = 0; i < equations.size(); ++i) {

final String A = equations.get(i).get(0);

final String B = equations.get(i).get(1);

graph.putIfAbsent(A, new HashMap<>());

graph.putIfAbsent(B, new HashMap<>());

graph.get(A).put(B, values[i]);

graph.get(B).put(A, 1.0 / values[i]);

}

for (int i = 0; i < queries.size(); ++i) {

final String A = queries.get(i).get(0);

final String C = queries.get(i).get(1);

if (!graph.containsKey(A) || !graph.containsKey(C))

ans[i] = -1.0;

else

ans[i] = divide(graph, A, C, new HashSet<>());

}

return ans;

}

private double divide(Map<String, Map<String, Double>> graph, final String A, final String C,

Set<String> seen) {

if (A.equals(C))

return 1.0;

seen.add(A);

for (final String B : graph.get(A).keySet()) {

if (seen.contains(B))

continue;

final double res = divide(graph, B, C, seen); // B / C

if (res > 0) // valid result

return graph.get(A).get(B) \* res; // A / C = (A / B) \* (B / C)

}

return -1.0; // invalid result

}

}