



# Huahua's Tech Road

October 16, 2018

Home Q&A Specials Algorithms ▾ Data Structure Difficulty ▾ Videos

LeetCode Donation

## DONTATION

如果您喜欢我们的内容，欢迎捐赠花花。

If you like my blog, donations are welcome.

Venmo 微信打赏



Paypal



## CATEGORIES

[Admin](#) (5)

[Array](#) (47)

[Bash](#) (1)

[Binary Search](#) (21)

[Bit](#) (12)

## 花花酱 LeetCode 399. Evaluate Division

BY ZXI ON DECEMBER 1, 2017

Post Views: 1,243

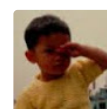


题目大意：给你一些含有变量名的分式的值，让你计算另外一些分式的值，如果不能计算返回-1。

### Problem:

Equations are given in the format  $A / B = K$ , where  $A$  and  $B$  are variables represented

## YOUTUBE CHANNEL



Hua Hua

YouTube 8K



优酷视频



大鱼号



## POPULAR POSTS

[花花酱 LeetCode 399. Evaluate Division](#)

609 views | posted on 12/01/2017

[LeetCode List](#)

[Contest](#) (1)  
[Data Structure](#) (14)  
[Desgin](#) (2)  
[Divide and conquer](#)  
(6)  
[Dynamic Programming](#) (68)  
[Easy](#) (24)  
[Geometry](#) (21)  
[Graph](#) (28)  
[Greedy](#) (23)  
[Hard](#) (17)  
[Hashtable](#) (54)  
[Heap](#) (4)  
[Leetcode](#) (21)  
[List](#) (19)  
[Math](#) (36)  
[Medium](#) (25)  
[Programming Language](#) (2)  
[Recursion](#) (3)  
[Search](#) (35)  
[Simulation](#) (27)  
[SP](#) (6)  
[SQL](#) (3)  
[Stack](#) (4)  
[String](#) (69)  
[Tree](#) (60)  
[Two pointers](#) (9)  
[Uncategorized](#) (5)  
[ZOJ](#) (3)

as strings, and  $k$  is a real number (floating point number). Given some queries, return the answers. If the answer does not exist, return `-1.0`.

### Example:

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]`.

The input is: `vector<pair<string, string>> equations`, `vector<double>& values`, `vector<pair<string, string>> queries`, where `equations.size() == values.size()`, and the values are positive. This represents the equations. Return `vector<double>`.

According to the example above:

```
1 equations = [ ["a", "b"], ["b", "c"] ],
2 values = [2.0, 3.0],
3 queries = [ ["a", "c"], ["b", "a"], ["a", "e"], ["a", "a"], ["x", "x"] ].
```

The input is always valid. You may assume that evaluating the queries will result in no division by zero and there is no contradiction.

479 views | posted on 12/11/2017

[花花酱 LeetCode 683. K Empty Slots](#)  
385 views | posted on 09/29/2017

[花花酱 LeetCode 681. Next Closest Time](#)  
374 views | posted on 09/24/2017

[花花酱 LeetCode 127. Word Ladder](#)  
370 views | posted on 09/25/2017

### TAGS

[array](#) [BFS](#) [binary search](#) [binary tree](#) [bit](#) [BST](#) [combination](#) [connected components](#) [conversion](#) [counting](#)

## DFS dp easy

[geometry](#) [graph](#)

[greedy](#) [hard](#)

[hashtable](#) [heap](#)

[list](#) [math](#) [matrix](#)

## medium

[merge](#) [Palindrome](#)

[permutation](#) [prefix sum](#)

[priority queue](#) [random](#)

[range](#) [recursion](#)

[reverse](#) [search](#)

[shortest path](#)

[simulation](#) [sort](#)

## Solution 1: Graph + DFS

$A / B = 2$   
 $g[A][B] = 2 \mid g[B][A] = 1.0 / 2$

Time complexity:  $O(e + q * e)$   
 Space complexity:  $O(e)$

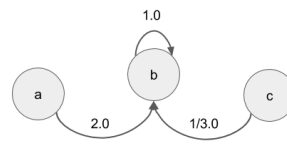
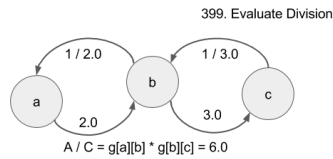
## Solution 2: Union Find

$A / B = 2 \rightarrow$  parents[A] = {B, 2}  
 parents[B] = {B, 1.0}

$B / C = 3 \rightarrow$  parents[C] = {B, 1.0/3}

$A / C ?$  p[A].key == p[C].key == B  
 $= p[A].val / p[C].val = 2 / (1.0/3) = 6$

Time complexity:  $O(e + q)$   
 Space complexity:  $O(e)$



<http://zxi.mytechroad.com/blog/>

## Solution 1: DFS

### C++

```

1 // Author: Huahua
2 // Runtime: 3 ms
3 class Solution {
4 public:
5     vector<double> calcEquation(vector
6     <pair<string, string>> equations, vect
7     or<double>& values, vector<pair<string
8     , string>> queries) {
9         // g[A][B] = k -> A / B = k
10        unordered_map<string, unordere
11        d_map<string, double>> g;
12        for (int i = 0; i < equations.
13        size(); ++i) {
14            const string& A = equation
15            s[i].first;
16            const string& B = equation
17            s[i].second;
18            const double k = values[i]
19            ;
20            g[A][B] = k;
21            g[B][A] = 1.0 / k;
22        }
23        vector<double> ans;
24        for (const auto& pair : querie
25        s) {
26            const string& X = pair.fir
27            st;
28            const string& Y = pair.sec
29            ond;
30            if (!g.count(X) || !g.coun
31            t(Y)) {
32                ans.push_back(-1.0);
33                continue;
34            }
35        }
36    }
37 };

```

[sorting stack](#)

[string subarray](#)

[subsequence sum](#)

[tree two pointers union](#)

[find](#)

**META**

[Register](#)

[Log in](#)

[Entries RSS](#)

[Comments RSS](#)

[WordPress.org](#)

**ARCHIVES**

[October 2018](#) (17)

[September](#)

[2018](#) (38)

[August 2018](#) (43)

[July 2018](#) (66)

[June 2018](#) (19)

[May 2018](#) (8)

[April 2018](#) (25)

[March 2018](#) (78)

[February 2018](#) (18)

[January 2018](#) (21)

[December](#)

[2017](#) (33)

[November](#)

[2017](#) (26)

[October 2017](#) (24)

[September](#)

[2017](#) (52)

```

34         unordered_set<string> visi
35 ted;
36         ans.push_back(divide(X, Y,
37 g, visited));
38     }
39     return ans;
40 }
41 private:
42     // get result of A / B
43     double divide(const string& A, con
44 st string& B,
45         unordered_map<string
, unordered_map<string, double>>& g,
        unordered_set<string
>& visited) {
        if (A == B) return 1.0;
        visited.insert(A);
        for (const auto& pair : g[A])
        {
            const string& C = pair.fir
st;
            if (visited.count(C)) cont
inue;
            double d = divide(C, B, g,
visited); // d = C / B
            // A / B = C / B * A / C
            if (d > 0) return d * g[A]
[C];
        }
        return -1.0;
    }
};

```

## Java

```

1 // Author: Huahua
2 // Running time: 74 ms
3 class Solution {
4     Map<String, HashMap<String, Double>>
g = new HashMap<>();
5
6     public double[] calcEquation(String[
][] equations, double[] values, String
[][] queries) {
7         for (int i = 0; i < equations.leng
th; ++i) {
8             String x = equations[i][0];
9             String y = equations[i][1];
10            double k = values[i];
11            g.computeIfAbsent(x, 1 -> new Ha
shMap<String, Double>()).put(y, k);
12            g.computeIfAbsent(y, 1 -> new Ha
shMap<String, Double>()).put(x, 1.0 /
k);
13        }
14
15        double[] ans = new double[queries.
length];

```

```

16
17     for (int i = 0; i < queries.length
; ++i) {
18         String x = queries[i][0];
19         String y = queries[i][1];
20         if (!g.containsKey(x) || !g.cont
ainsKey(y)) {
21             ans[i] = -1.0;
22         } else {
23             ans[i] = divide(x, y, new Hash
Set<String>());
24         }
25     }
26
27     return ans;
28 }
29
30 private double divide(String x, Stri
ng y, Set<String> visited) {
31     if (x.equals(y)) return 1.0;
32     visited.add(x);
33     if (!g.containsKey(x)) return -1.0
;
34     for (String n : g.get(x).keySet())
{
35         if (visited.contains(n)) continu
e;
36         visited.add(n);
37         double d = divide(n, y, visited)
;
38         if (d > 0) return d * g.get(x).g
et(n);
39     }
40     return -1.0;
41 }
42 }

```

## Python3

```

1  """
2  Author: Huahua
3  Running time: 32 ms (beats 100%)
4  """
5  class Solution:
6      def calcEquation(self, equations, va
lues, queries):
7          def divide(x, y, visited):
8              if x == y: return 1.0
9              visited.add(x)
10             for n in g[x]:
11                 if n in visited: continue
12                 visited.add(n)
13                 d = divide(n, y, visited)
14                 if d > 0: return d * g[x][n]
15             return -1.0
16
17         g = collections.defaultdict(dict)
18         for (x, y), v in zip(equations, va

```

```

    lues):
19         g[x][y] = v
20         g[y][x] = 1.0 / v
21
22         ans = [divide(x, y, set()) if x in
                g and y in g else -1 for x, y in queries]
23         return ans

```

## Solution 2: Union Find

### C++

```

1  // Author: Huahua
2  // Runtime: 3 ms
3  class Solution {
4  public:
5      vector<double> calcEquation(const vector<pair<string, string>>& equations,
                                   vector<double>& values, const vector<pair<string, string>>& queries) {
6          // parents["A"] = {"B", 2.0} -> A = 2.0 * B
7          // parents["B"] = {"C", 3.0} -> B = 3.0 * C
8          unordered_map<string, pair<string, double>> parents;
9
10         for (int i = 0; i < equations.size(); ++i) {
11             const string& A = equations[i].first;
12             const string& B = equations[i].second;
13             const double k = values[i];
14             // Neighbor is in the forest
15             if (!parents.count(A) && !parents.count(B)) {
16                 parents[A] = {B, k};
17                 parents[B] = {A, 1.0 / k};
18             } else if (!parents.count(A)) {
19                 parents[A] = {B, k};
20             } else if (!parents.count(B)) {
21                 parents[B] = {A, 1.0 / k};
22             } else {
23                 auto& rA = find(A, parents);
24                 auto& rB = find(B, parents);
25                 parents[rA.first] = {rB.first, k / rA.second * rB.second};
26             }
27         }
28         vector<double> ans;

```

```

40     for (const auto& pair : queries) {
41         const string& X = pair.first;
42         const string& Y = pair.second;
43         if (!parents.count(X) || !parents.
44 count(Y)) {
45             ans.push_back(-1.0);
46             continue;
47         }
48         auto& rX = find(X, parents); // {r
X, X / rX}
49         auto& rY = find(Y, parents); // {r
Y, Y / rY}
50         if (rX.first != rY.first)
51             ans.push_back(-1.0);
52         else // X / Y = (X / rX / (Y / rY)
53 )
54             ans.push_back(rX.second / rY.sec
ond);
55     }
56     return ans;
57 }
58 private:
59 pair<string, double>& find(const str
ing& C, unordered_map<string, pair<str
ing, double>>& parents) {
60     if (C != parents[C].first) {
61         const auto& p = find(parents[C].
first, parents);
62         parents[C].first = p.first;
63         parents[C].second *= p.second;
64     }
65     return parents[C];
66 }
67 };

```

## Java

```

1  // Author: Huahua
2  // Running time: 3 ms
3  class Solution {
4      class Node {
5          public String parent;
6          public double ratio;
7          public Node(String parent, double
8 ratio) {
9              this.parent = parent;
10             this.ratio = ratio;
11         }
12     }
13
14     class UnionFindSet {
15         private Map<String, Node> parents
16 = new HashMap<>();
17
18         public Node find(String s) {
19             if (!parents.containsKey(s)) ret
20 urn null;
21             Node n = parents.get(s);

```

```

22         if (!n.parent.equals(s)) {
23             Node p = find(n.parent);
24             n.parent = p.parent;
25             n.ratio *= p.ratio;
26         }
27         return n;
28     }
29
30     public void union(String s, String
31 p, double ratio) {
32         boolean hasS = parents.containsKey(s);
33         boolean hasP = parents.containsKey(p);
34         if (!hasS && !hasP) {
35             parents.put(s, new Node(p, ratio));
36             parents.put(p, new Node(p, 1.0));
37         } else if (!hasP) {
38             parents.put(p, new Node(s, 1.0 / ratio));
39         } else if (!hasS) {
40             parents.put(s, new Node(p, ratio));
41         } else {
42             Node rS = find(s);
43             Node rP = find(p);
44             rS.parent = rP.parent;
45             rS.ratio = ratio / rS.ratio *
46 rP.ratio;
47         }
48     }
49
50     public double[] calcEquation(String[
51 ][ ] equations, double[] values, String
52 [ ][ ] queries) {
53         UnionFindSet u = new UnionFindSet(
54 );
55
56         for (int i = 0; i < equations.length; ++i)
57             u.union(equations[i][0], equations[i][1], values[i]);
58
59         double[] ans = new double[queries.length];
60
61         for (int i = 0; i < queries.length; ++i) {
62             Node rx = u.find(queries[i][0]);
63             Node ry = u.find(queries[i][1]);
64             if (rx == null || ry == null || !rx.parent.equals(ry.parent))
65                 ans[i] = -1.0;
66             else
67                 ans[i] = rx.ratio / ry.ratio;
68         }
69
70         return ans;

```



```
}  
}
```

## Python3

```
1  """  
2  Author: Huahua  
3  Running time: 32 ms (beats 100%)  
4  """  
5  class Solution:  
6      def calcEquation(self, equations, values, queries):  
7          def find(x):  
8              if x != U[x][0]:  
9                  px, pv = find(U[x][0])  
10                 U[x] = (px, U[x][1] * pv)  
11                 return U[x]  
12  
13             def divide(x, y):  
14                 rx, vx = find(x)  
15                 ry, vy = find(y)  
16                 if rx != ry: return -1.0  
17                 return vx / vy  
18  
19             U = {}  
20             for (x, y), v in zip(equations, values):  
21                 if x not in U and y not in U:  
22                     U[x] = (y, v)  
23                     U[y] = (x, 1.0 / v)  
24                 elif x not in U:  
25                     U[x] = (y, v)  
26                 elif y not in U:  
27                     U[y] = (x, 1.0 / v)  
28                 else:  
29                     rx, vx = find(x)  
30                     ry, vy = find(y)  
31                     U[rx] = (ry, v / vx * vy)  
32  
33             ans = [divide(x, y) if x in U and y in U else -1 for x, y in queries]  
34             return ans
```

### Related Problems:

- [\[解题报告\] LeetCode 684. Redundant Connection](#)
- [\[解题报告\] LeetCode 547. Friend Circles](#)
- [\[解题报告\] LeetCode 737. Sentence Similarity II](#)

请尊重作者的劳动成果，转载请注明出处！花花保留对文章 / 视频的所有权利。  
如果您喜欢这篇文章 / 视频，欢迎您捐赠花花。  
If you like my articles / videos, donations are welcome.



Published in [Graph](#)

DFS   graph   hard   union find



**zxi**

More from **Graph**

[More posts in Graph »](#)

花花酱 LeetCode  
924. Minimize  
Malware Spread

花花酱 LeetCode  
886. Possible  
Bipartition

花花酱 LeetCode  
882. Reachable  
Nodes In  
Subdivided Graph

花花酱 LeetCode  
785. Is Graph  
Bipartite?

Be First to Comment

## Leave a Reply

You must be [logged in](#) to post a comment.

---

# Huahua's Tech Road



**Mission News Theme** by Compete Themes.