Practice Compete Jobs Rank Leaderboard

Dashboard > Algorithms > Graph Theory > Coprime Paths

Badge Progress (Details)

Points: 126 Rank: 270095

Coprime Paths

by VastoLorde95

Problem Submissions Leaderboard Discussions Editorial

You are given an undirected, connected graph, G, with n nodes and m edges where $m = n - 1$. Each node i is initially assigned a value, $node_i$, that has *at most* 3 prime divisors.

You must answer q queries in the form `u v`. For each query, find and print the *number of* (x, y) *pairs* of nodes on the path between u and v such that $gcd(node_x, node_y) = 1$ and the length of the path between u and v is minimal among all paths from u to v.

Input Format

The first line contains two space-separated integers describing the respective values of n and q.
The second line contains n space-separated integers describing the respective values of $node_1, node_2, \ldots, node_n$.
Each of the $n - 1$ subsequent lines contains two space-separated integers, u and v, describing an edge between nodes u and v.
Each of the q subsequent lines contains two space-separated integers, u and v, describing a query.

Constraints

- $1 \leq n, q \leq 25 \times 10^3$
- $1 \leq node_i \leq 10^7$
- $1 \leq u, v \leq n$

Output Format

For each query, print an integer on a new line denoting the *number of* (x, y) *pairs* of nodes on the path between u and v such that $gcd(node_x, node_y) = 1$ and the length of the path between u and v is minimal among all paths from u to v.

Sample Input 0

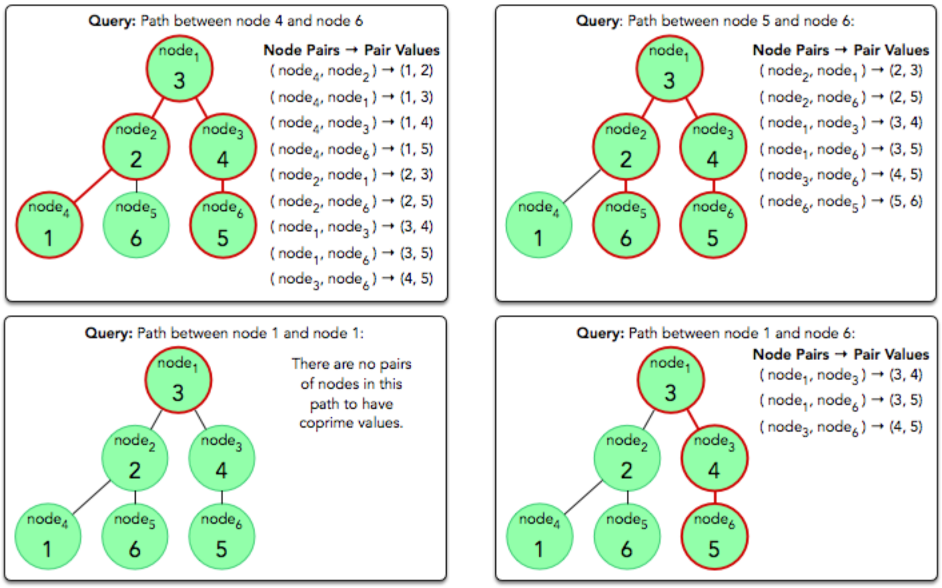
```
6 5
3 2 4 1 6 5
1 2
1 3
2 4
2 5
3 6
4 6
5 6
1 1
1 6
6 1
```

Sample Output 0

```
9
6
0
3
3
```

Explanation 0

The diagram below depicts graph G and the $u \leftrightarrow v$ paths specified by each query, as well as the *Pair Values* for each path in the form $(node_x, node_y)$:


Query: Path between node 4 and node 6
node$_1$ 3
node$_2$ 2
node$_3$ 4
node$_4$ 1
node$_5$ 6
node$_6$ 5
Node Pairs → Pair Values
$(node_4, node_2) \rightarrow (1, 2)$
$(node_4, node_1) \rightarrow (1, 3)$
$(node_4, node_3) \rightarrow (1, 4)$
$(node_4, node_6) \rightarrow (1, 5)$
$(node_2, node_1) \rightarrow (2, 3)$
$(node_2, node_6) \rightarrow (2, 5)$
$(node_1, node_3) \rightarrow (3, 4)$
$(node_1, node_6) \rightarrow (3, 5)$
$(node_3, node_6) \rightarrow (4, 5)$
Query: Path between node 5 and node 6:
Node Pairs → Pair Values
$(node_2, node_1) \rightarrow (2, 3)$
$(node_2, node_6) \rightarrow (2, 5)$
$(node_1, node_3) \rightarrow (3, 4)$
$(node_1, node_6) \rightarrow (3, 5)$
$(node_3, node_6) \rightarrow (4, 5)$
$(node_6, node_5) \rightarrow (5, 6)$
Query: Path between node 1 and node 1:
There are no pairs of nodes in this path to have coprime values.
Query: Path between node 1 and node 6:
Node Pairs → Pair Values
$(node_1, node_3) \rightarrow (3, 4)$
$(node_1, node_6) \rightarrow (3, 5)$
$(node_3, node_6) \rightarrow (4, 5)$

Recall that, for each queried path, we want to find and print the number of (x, y) pairs of nodes such that $gcd(node_x, node_y) = 1$.

Submissions:69

Max Score:80

Difficulty: Expert

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int q = in.nextInt();
        int[] nodes = new int[n];
        for(int nodes_i=0; nodes_i < n; nodes_i++){
            nodes[nodes_i] = in.nextInt();
        }
        int[][] edges = new int[n-1][2];
        for(int edges_i=0; edges_i < n-1; edges_i++){
            for(int edges_j=0; edges_j < 2; edges_j++){
                edges[edges_i][edges_j] = in.nextInt();
            }
        }
        for(int a0 = 0; a0 < q; a0++){
            int u = in.nextInt();
            int v = in.nextInt();
            // your code goes here
        }
    }
}
```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature