Question 2 Max. score: 100.00

Minimize difference

You are given a tree with N vertices numbered from 1 to N. The i^{th} edge connects Vertex x_i and Vertex y_i bidirectionally. You have to divide this tree into *three* connected components by cutting any two edges of the tree. Let the three components be C_1, C_2, and C_3. Let X_1, X_2, and X_3 be the *XOR* of all the vertices of the components C_1, C_2, and C_3 respectively.

Task

Minimize the difference between the maximum and minimum xor values of the components. In short, you have to minimize the value of $max(X_1, X_2, X_3) - min(X_1, X_2, X_3)$.

Notes

- A *tree* is an undirected, connected and acyclic graph.
- A set of nodes forms a *connected component* in a tree if any node from the set of nodes can reach any other node by traversing edges.
- The bitwise *XOR* of integers *A* and *B*, *A XOR B*, is when *A XOR B* is written in base two, the digit in the 2^ks place ($k \geq 0$) is 1 if exactly one of A and B is 1, and 0 otherwise. For example, *XOR* of $(101)_2$ and $(110)_2$ is $(011)_2$
- *Cutting an edge* means partitioning the vertices of the tree into two disjoint subsets. In other words, cutting an edge results in an increase in the number of connected components.

Example

Assumptions

- $N = 4$
- *edges = [[1, 2], [2, 3], [4, 1]]*

Approach

- You can cut the edge between *2 and 3* and the edge between *1 and 4*.
- The components are now $C_1 = \{1, 2\}$, $C_2 = \{3\}$ and $C_3 = \{4\}$. So the *XOR* values are $X_1 = 3 (1 \oplus 2)$, $X_2 = 3$ and $X_3 = 4$.
- Thus, the answer is *4 - 3 = 1*.

Function description

Complete the function *solve* provided in the editor. This function takes the following *2* parameters and returns the minimum difference between the maximum and minimum *XOR* values of the components:

- *N:* Represents the number of vertices
- *edges:* Represents the array containing edges of the tree

Input format

Note: This is the input format that you must use to provide custom input (available above the **Compile and Test** button).

- The first line contains T denoting the number of test cases. T also specifies the number of times you have to run the *solve* function on a different set of inputs.
- For each test case:
  - The first line contains integer N denoting the number of vertices.
  - Each of the following $N - 1$ line contains two integers x and y which indicates that there is an edge between x and y.

Output format

For each test case in a new line, print the minimum possible difference between the maximum and minimum *XOR* values of the components.

Constraints

$1 \leq T \leq 10$

$3 \leq N \leq 3000$

$1 \leq x, y \leq N$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

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

Sample output

```
3
1
```

View more

Explanation

The first line contains the number of test cases, $T = 2$.

For the first test case

Given

- *N = 5*
- *edges = [[1, 2], [1, 4], [1, 5], [3, 4]]*

Approach

- You can cut the edge between *3 and 4* and the edge between *1 and 4*.
- The components are now C_1 = {1, 2, 5}, C_2 = {3} and C_3 = {4}. So the *XOR* values are X_1 = 6(1⊕2⊕5), X_2 = 3 and X_3 = 4.
- Thus, the answer is *6 - 3 = 3*.

For the second test case

- It is the same as the example. Please refer to that.

The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Sample input 1

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

Sample output 1

```
0
0
2
```

View more

Sample input 2

```
3
4
3 4
1 3
4 2
4
3 2
4 1
3 1
9
```

Sample output 2

```
2
2
1
```

View more

Note:

Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Limits

Time Limit: 1.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB

Scoring

Score is assigned if any testcase passes

Allowed Languages

Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic

New Submission | All Submissions

Auto-complete ready! Save Java 8 (openjdk 1.8.0_241)

```
            graph[i] = new HashSet<Integer>();
        }

        for(int[] edge: edges) {
            graph[edge[0]].add(edge[1]);
            graph[edge[1]].add(edge[0]);
        }

        for(int i=0; i<edges.length-1; i++) {
            for(int j=i+1; j<edges.length; j++) {
                graph[edges[i][0]].remove(edges[i][1]);
                graph[edges[i][1]].remove(edges[i][0]);
                graph[edges[j][0]].remove(edges[j][1]);
                graph[edges[j][1]].remove(edges[j][0]);

                int xors[] = bfs(graph);
                res = Math.min(res, max(xors[0], xors[1], xors[2]) - min(xors[0], xors[1], xors[2]));

                graph[edges[i][0]].add(edges[i][1]);
                graph[edges[i][1]].add(edges[i][0]);
                graph[edges[j][0]].add(edges[j][1]);
```

101:2 vscode

Test against custom input ▾ Compile & Test code Submit code

Custom input populated

Submission ID: 71224333 / 9 seconds ago

RESULT: Partially accepted

Refer judge environment

Time (sec)	Memory (KiB)	Language
10.40387	86400	Java 8

Visibility	Input	Result	Time (sec)	Memory (KiB)	Your Output	Correct Output
	Input #1	Accepted	0.076984	82992		
	Input #2	Accepted	0.069173	80936		
	Input #3	Accepted	0.068537	80252	-	-
	Input #4	Accepted	0.0963	81572	-	-
	Input #5	Accepted	0.085373	81604	-	-
	Input #6	Time limit exceeded	2.0015	86276	-	-
	Input #7	Time limit exceeded	2.0015	84084	-	-
	Input #8	Time limit exceeded	2.0015	86208	-	-
	Input #9	Time limit exceeded	2.0015	86400	-	-
	Input #10	Time limit exceeded	2.0015	83856	-	-

‹ Previous Question

Next Question ›