Redundant Connection

Try to solve the Redundant Connection problem.

Statement

We're given an undirected graph consisting of n nodes. The graph is represented as an array called `edges`, of length n, where `edges[i] = [a, b]` indicates that there is an edge between nodes `a` and `b` in the graph.

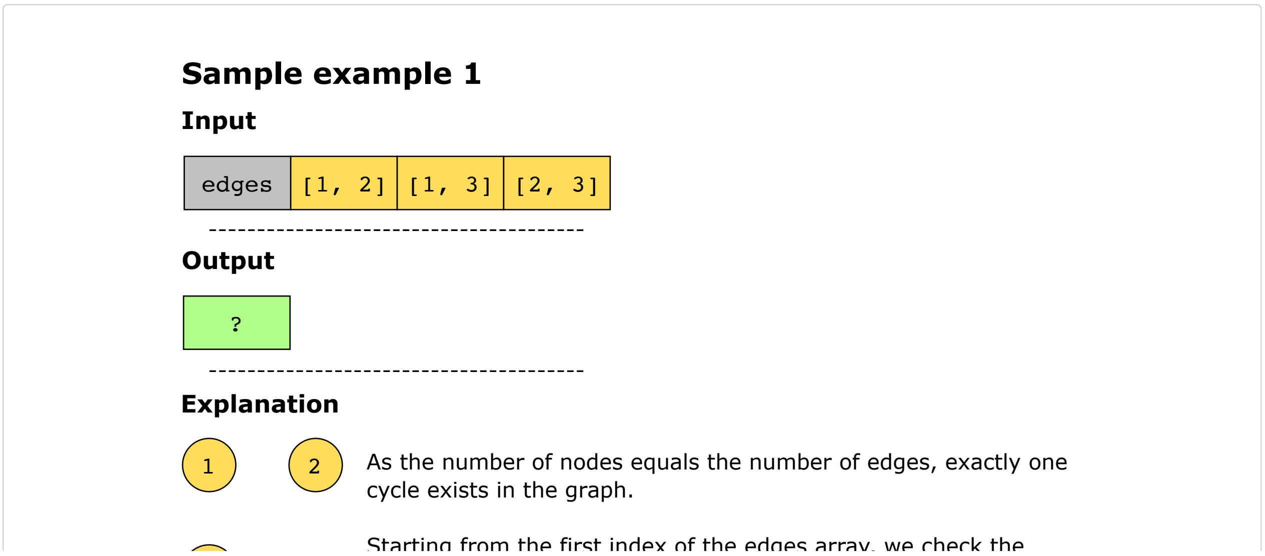
Return an edge that can be removed to make the graph a tree of n nodes. If there are multiple candidates for removal, return the edge that occurs last in `edges`.

Constraints:

- $3 \leq n \leq 1000$
- `edges.length`$= n$
- `edges[i].length` $= 2$
- $1 \leq$ `a` $<$ `b` $\leq n$
- $a \neq b$
- There are no repeated edges.
- The given graph is connected.
- The graph contains only one cycle.

Example

We can understand the problem more comprehensively by looking at the illustration below:



Sample example 1

Input

edges	[1, 2]	[1, 3]	[2, 3]

Output

?

Explanation

1 2 As the number of nodes equals the number of edges, exactly one cycle exists in the graph.

Starting from the first index of the edges array, we check the

Starting from the first index of the edges array, we check the connectivity of the nodes.

1 of 5

Sample example 1

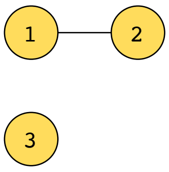
Input

edges	[1, 2]	[1, 3]	[2, 3]

Output

?

Explanation



Edge **[1, 2]**: We connect nodes **1** and **2** with an undirected edge.

2 of 5

Sample example 1

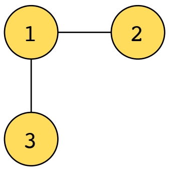
Input

edges	[1, 2]	[1, 3]	[2, 3]

Output

?

Explanation



Edge **[1, 3]**: We connect nodes **1** and **3** with an undirected edge.

3 of 5

Sample example 1

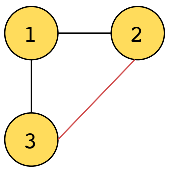
Input

edges	[1, 2]	[1, 3]	[2, 3]

Output

?

Explanation



Since all three nodes are already connected, connecting nodes **2** and **3** will yield a redundant edge, forming a cycle.

4 of 5

Sample example 1

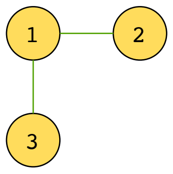
Input

edges	[1, 2]	[1, 3]	[2, 3]

Output

[2, 3]

Explanation



As **[2, 3]** was the redundant edge, we remove it, making the number of the edges in our graph one less than the number of nodes. We return the edge that we removed, that is: **[2, 3]**.

5 of 5

Test your understanding of the problem

Let's take a moment to make sure we have correctly understood the problem. The quiz below helps us to check that we are solving precisely the right problem:

Redundant Connection

1 What is the output if the following array of edges is provided as input?

edges = [[1, 2], [2, 3], [3, 4], [1, 4], [1, 5]]

A) [2, 3]

B) [3, 4]

C) [1, 4]

D) [1, 5]

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play: re-arrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to re-arrange them in the correct sequence.

Traverse the `edges` array from the first index to the last index.

For each edge, connect the two nodes by marking them. This makes them part of a single connected component.

If the current edge connects two nodes that are already marked as part of the connected component, the edge is redundant, so we return it.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in `RedundantConnections.java` in the following coding playground. You will need the provided supporting code to implement your solution.

> **Note:** To solve this problem, you may need to change the implementation of the functions in the given Union Find class.

Java

RedundantConnections.java

UnionFind.java

```java
import java.util.*;

public class RedundantConnections{
    public static int[] redundantConnection(int[][] edges) {

        // Your code will replace this placeholder return statement
        return new int[]{};
    }
}
```

Powered by AI | Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

```
[[1,2],[1,3],[2,3]]
```

Redundant Connection

← Back

Union Find: Introduction

Next →

Solution: Redundant C...

✓ Mark as Completed