Common Reachable Node

Question 879 of 1031

Medium

You are given a two-dimensional list of integers `edges` representing a directed graph and integers `a` and `b`. Each element in `edges` contains `[u, v]` meaning there's an edge from `u` to `v`.

Return whether there's some node `c` such that we can go from `c` to `a` and `c` to `b`.

Constraints

- `1 ≤ n ≤ 100,000` where `n` is the length of `edges`

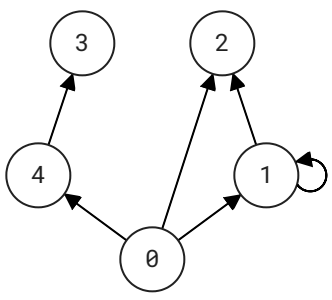
```
1 import java.util.*;
2
3 class Solution {
4     public boolean solve(int[][] edges, int a,
  int b) {
5
6     }
7 }
```

Example 1

Input

Visualize

```
edges =
```




```
a = 2
b = 3
```

Output

```
true
```

Explanation

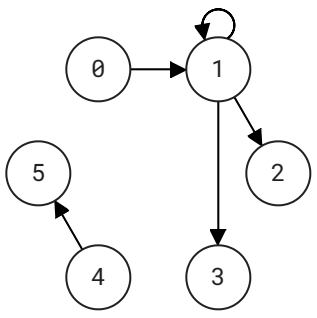
We can reach `2` and `3` from `0`

Example 2

Input

Visualize

```
edges =
```




```
a = 2
b = 4
```

Output

```
false
```

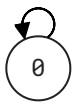
Explanation

`2` and `4` are on different connected components.

Example 3

Input

Visualize

edges = 


```
a = 0
b = 0
```

Output

Java ^ Run ▶ Submit