# 178 · Graph Valid Tree

Algorithms Medium Accepted Rate 35%



## Description

Given `n` nodes labeled from `0` to `n - 1` and a list of `undirected` edges (each edge is a pair of nodes), write a function to check whether these edges make up a valid tree.

You can assume that no duplicate edges will appear in edges. Since all edges are `undirected`, `[0, 1]` is the same as `[1, 0]` and thus will not appear together in edges.

## Example

**Example 1:**

```
Input: n = 5 edges = [[0, 1], [0, 2], [0, 3], [1, 4]]
Output: true.
```

**Example 2:**

```
Input: n = 5 edges = [[0, 1], [1, 2], [2, 3], [1, 3], [1, 4]]
Output: false.
```

## Tags

## Related Problems

| | | | |
|---|---|---|---|
| {A} | 431 | Connected Component in Undirected Graph | Medium |
| {A} | 589 | Connecting Graph | Medium |
| {A} | 750 | Portal | Medium |
| {A} | 814 | Shortest Path in Undirected Graph | Medium |

C++ autocomplete tu

```
    }
    bool validTree
        //write y
        vector<vec
        for(auto i
            int a=
            int b=
            adj[a]
            adj[b]
        }
        int count=
        vector<boo
        for(int i=
            if(vis
                co
                if
                {
                }
            }
        }
        if(count>1
            re
```

Test Data Commit Output

Accepted

81 ms time cost · 3.47 MB

view ranking distribution contribu

Share your acceptance to your

Console