

[Description](#)
[Solution](#)
[Discuss \(243\)](#)
[Submissions](#)

## 1245. Tree Diameter

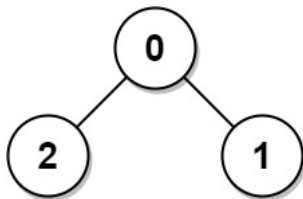
**Medium**
 700
 17

The **diameter** of a tree is the **number of edges** in the longest path in that tree.

There is an undirected tree of  $n$  nodes labeled from  $0$  to  $n - 1$ . You are given a 2D array `edges` where `edges.length == n - 1` and `edges[i] = [ai, bi]` indicates that there is an undirected edge between nodes  $a_i$  and  $b_i$  in the tree.

Return the **diameter** of the tree.

Example 1:

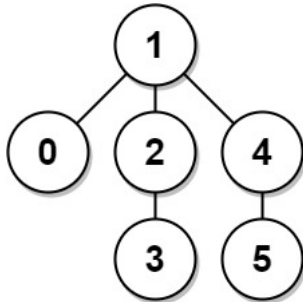


**Input:** `edges = [[0,1],[0,2]]`

**Output:** 2

**Explanation:** The longest path of the tree is the path  $1 - 0 - 2$ .

Example 2:



**Input:** `edges = [[0,1],[1,2],[2,3],[1,4],[4,5]]`

**Output:** 4

**Explanation:** The longest path of the tree is the path  $3 - 2 - 1 - 4 - 5$ .

Constraints:

- $n == \text{edges.length} + 1$
- $1 \leq n \leq 10^4$
- $0 \leq a_i, b_i < n$
- $a_i \neq b_i$

Accepted 29,235 Submissions 47,513

Seen this question in a real interview before?

Companies [i](#)

Related Topics

*i* C++ Autocomplete

```

1 class Solution {
2 public:
3     int diameter(vector<int> adj[],int par,int &maxLen,int
parent)
4     {
5         int maxi = 0,secondMaxi = 0;
6         for(int children : adj[par])
7         {
8             if(parent == children)
9                 continue;
10            int len = diameter(adj,children,maxLen,par);
11            if(len>maxi)
12            {
13                secondMaxi = maxi;
14                maxi = len;
15            }
16            else if(len>secondMaxi)
17            {
18                secondMaxi = len;
19            }
20        }
21        maxLen = max(maxLen,maxi+secondMaxi);
22        return maxi+1;
23    }
24
25    int treeDiameter(vector<vector<int>>& edges) {
26        int n = edges.size();
27
28        vector<int> adj[n+1];
29
30        for(int i=0;i<edges.size();i++)
31        {
32            adj[edges[i][0]].push_back(edges[i][1]);
33            adj[edges[i][1]].push_back(edges[i][0]);
34        }
35
36        int maxLen = 0;
37        diameter(adj,0,maxLen,-1);
38        return maxLen;
39    }
40 };
41
42 /*
43 m-2 this is bfs solution good technique that one should
know nice solution
44
45 class Solution {
46     public int treeDiameter(int[][] edges) {
47
48         List<Set<Integer>> graph = new
ArrayList<Set<Integer>>();
49         for (int i = 0; i < edges.length + 1; ++i) {
50             graph.add(new HashSet<Integer>());
51         }
52     }

```

NEW

Testcase Run Code Result Debugger

Accepted

Runtime: 0 ms

Your input

`[[0,1],[1,2],[2,3],[1,4],[4,5]]`

Output

4

Expected

4

Console...

[Use Example Testcases](#)

Run Code ^

Submit

Problems

Pick One

< Prev

1245/2509

Next >

