# 5702. Find Center of Star Graph

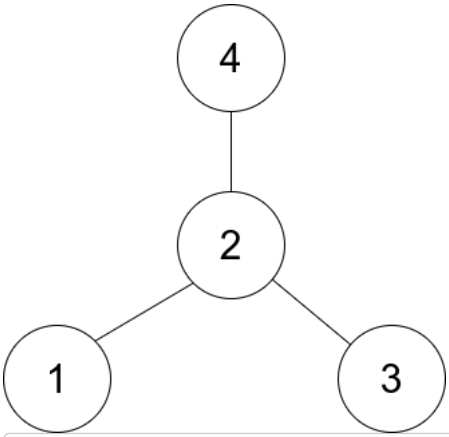
My Submissions (/contest/weekly-contest-232/problems/find-center-of-star-graph/submissions/) Back to Contest (/contest/weekly-contest-232/)

There is an undirected **star** graph consisting of `n` nodes labeled from `1` to `n`. A star graph is a graph where there is one **center** node and **exactly** `n - 1` edges that connect the center node with every other node.

You are given a 2D integer array `edges` where each `edges[i] = [u_i, v_i]` indicates that there is an edge between the nodes `u_i` and `v_i`. Return the center of the given star graph.

| | |
|---|---|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Medium |

**Example 1:**




```
Input: edges = [[1,2],[2,3],[4,2]]
Output: 2
Explanation: As shown in the figure above, node 2 is connected to every other node, so 2 is the center.
```

**Example 2:**

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

**Constraints:**

- `3 <= n <= 10^5`
- `edges.length == n - 1`
- `edges[i].length == 2`
- `1 <= u_i, v_i <= n`
- `u_i != v_i`
- The given `edges` represent a valid star graph.

JavaScript

```javascript
/**
 * @param {number[][]} edges
 * @return {number}
 */
const findCenter = (edges) => {
    let m = new Map();
    for (const e of edges) {
        m.set(e[0], m.get(e[0]) + 1 || 1);
        m.set(e[1], m.get(e[1]) + 1 || 1);
    }
    m = sortMapByValue(m);
    return m.keys().next().value;
};

const sortMapByValue = (map) => {
    return new Map([...map].sort((a, b) => b[1] - a[1]));
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/467469040/) More Details (/submissions/detail/467469040/)

Share your acceptance!