Minimum Height Trees

Got it! ✅ Let's explain this step by step in very simple terms.

This problem is about finding the Minimum Height Tree Roots of a graph (basically finding the *centers* of the tree).

✅ Problem Setup

We have a tree with `n` nodes (0 to n-1) and edges.
We want to find which nodes can be the root of a Minimum Height Tree.

For this example:

lua Copy Edit

```lua
n = 4
edges = [[1,0], [1,2], [1,3]]
```

This means:

markdown Copy Edit

```
  0
   \
    1
   / \
  2   3
```

So node 1 connects to 0,2,3.

✅ Step 1: Build Adjacency List

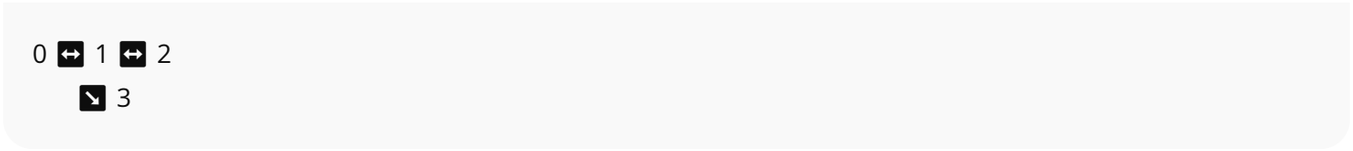
We build a list of neighbors for each node:

bash Copy Edit

```bash
graph[0] = {1}       # 0 connects to 1
graph[1] = {0,2,3}   # 1 connects to 0,2,3
graph[2] = {1}       # 2 connects to 1
graph[3] = {1}       # 3 connects to 1
```

So:

markdown Copy Edit



✅ Step 2: Find Initial Leaves

A leaf is a node with only 1 connection.

Here:

- Node `0` has only 1 neighbor → it's a leaf
- Node `2` has only 1 neighbor → it's a leaf
- Node `3` has only 1 neighbor → it's a leaf
- Node `1` has 3 neighbors → NOT a leaf

So initial leaves = [0, 2, 3]

✅ Step 3: Remove Leaves Layer by Layer

Think of it like peeling an onion → remove outer leaves layer by layer until only the center remains.

👉 Remove leaf 0 → Node `1` loses a neighbor
👉 Remove leaf 2 → Node `1` loses another neighbor
👉 Remove leaf 3 → Node `1` loses another neighbor

Now Node `1` has no neighbors left → it's the center of the tree.

✅ Final Result

Only node `1` remains after peeling → so the Minimum Height Tree root is [1].

✅ Why does this work?

- A Minimum Height Tree root must be at the center of the tree.
- If you peel off all leaves layer by layer, you'll always end up with 1 or 2 center nodes.

In this example, after peeling the outer layer, only `1` remains → that's the center.

Would you like me to:

✅ Show another example with 2 final centers?

✅ Or visualize it step by step with a layer peeling diagram?