Let's walk through the **recursion stack** for the problem with a simple example of a binary tree:

The longest path (diameter) in this tree is 4→2→1→3 and its length is 3 edges.

Step-by-Step Recursion Walkthrough

The recursive function `dfs(node)` computes:

1. The height of the left and right subtrees of the node.
2. Updates `self.res` to track the diameter at each node.
3. Returns the height of the subtree rooted at `node`.

Function Calls and Return Values

We start at the root node (`1`) and recursively compute heights for the left and right subtrees.

1. Call Stack Visualization for `dfs(1)`:

- **Current Node**: 1
- We first compute the left subtree (`dfs(2)`).

---

2. Call Stack Visualization for `dfs(2)`:

- **Current Node**: 2
- We compute the left subtree (`dfs(4)`).

---

3. Call Stack Visualization for `dfs(4)`:

- **Current Node**: 4

- No left or right children: both `dfs(None)` return 0.
- **Update**:
  - `left = 0`, `right = 0`
  - `self.res = max(self.res, left + right) = max(0, 0 + 0) = 0`
- **Return Height**: 1+max(0,0)=1.

---

4. Back to `dfs(2)`: Compute `dfs(5)`:

- **Current Node**: 5
- No left or right children: both `dfs(None)` return 0.
- **Update**:
  - `left = 0`, `right = 0`
  - `self.res = max(self.res, left + right) = max(0, 0 + 0) = 0`
- **Return Height**: 1+max(0,0)=1.

---

5. Back to `dfs(2)`:

- **Left and Right Heights**:
  - `left = 1` (from `dfs(4)`), `right = 1` (from `dfs(5)`).
- **Update**:
  - `self.res = max(self.res, left + right) = max(0, 1 + 1) = 2`
- **Return Height**: 1+max(1,1)=2.

---

6. Back to `dfs(1)`: Compute `dfs(3)`:

- **Current Node**: 3
- No left or right children: both `dfs(None)` return 0.
- **Update**:
  - `left = 0`, `right = 0`
  - `self.res = max(self.res, left + right) = max(2, 0 + 0) = 2`
- **Return Height**: 1+max(0,0)=1.

---

7. Back to `dfs(1)`:

- **Left and Right Heights**:
  - `left = 2` (from `dfs(2)`), `right = 1` (from `dfs(3)`).

- **Update**:
    - `self.res = max(self.res, left + right) = max(2, 2 + 1) = 3`
- **Return Height**: 1+max(2,1)=3

Final Results

- `self.res = 3`: The diameter of the tree (longest path).
- Return `self.res` from `diameterOfBinaryTree`.