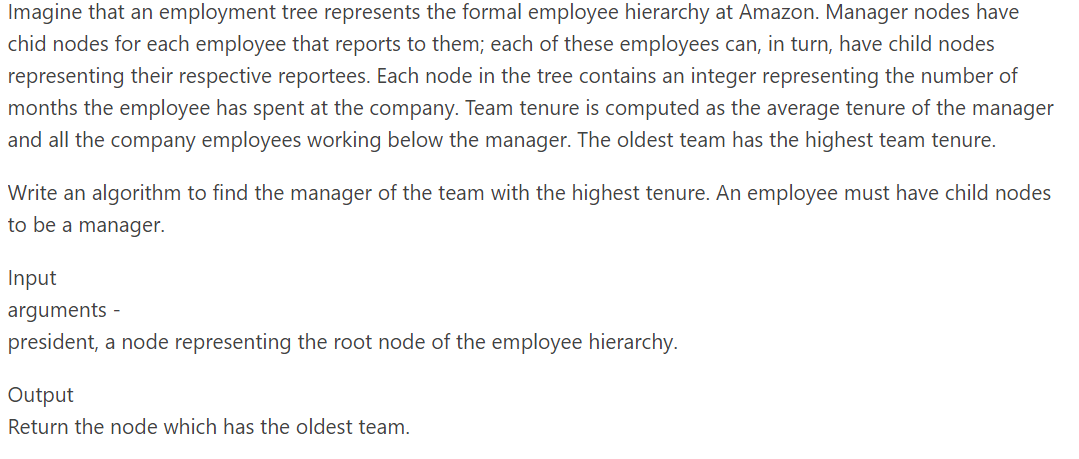
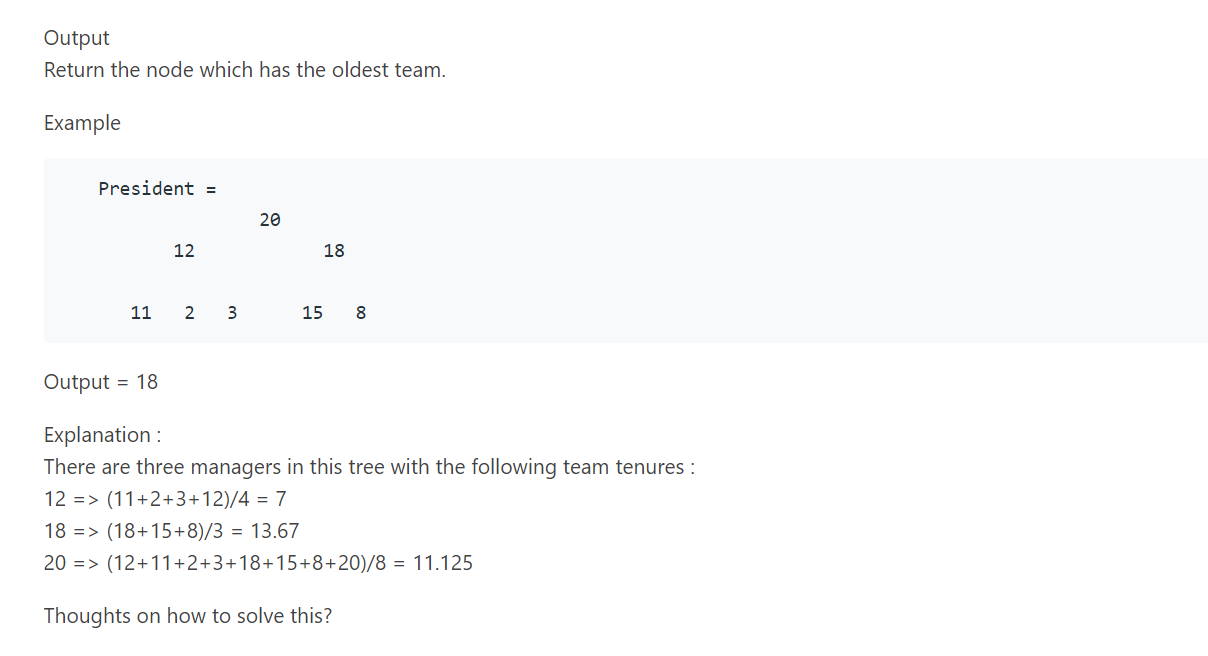
## Amazon 2020| Oldest team || **Amazon OA - Highest Tenure**

LEETCODE : <https://leetcode.com/discuss/interview-question/826071/amazon-2020-oldest-team>





class Solution {

// for each node in the tree, we will maintain three values

class State {

// count of nodes in the subtree

int nodeCount;

// sum of values in the subtree

int valueSum;

// max average found in the subtree

double maxAverage;

State(int nodes, int sum, double maxAverage) {

this.nodeCount = nodes;

this.valueSum = sum;

this.maxAverage = maxAverage;

}

}

public double maximumAverageSubtree(TreeNode root) {

return maxAverage(root).maxAverage;

}

State maxAverage(TreeNode root) {

if (root == null) {

return new State(0, 0, 0);

}

// postorder traversal, solve for both child nodes first.

State left = maxAverage(root.left);

State right = maxAverage(root.right);

// now find nodeCount, valueSum and maxAverage for current node `root`

int nodeCount = left.nodeCount + right.nodeCount + 1;

int sum = left.valueSum + right.valueSum + root.val;

double maxAverage = Math.max(

(1.0 \* (sum)) / nodeCount, // average for current node

Math.max(right.maxAverage, left.maxAverage) // max average from child nodes

);

return new State(nodeCount, sum, maxAverage);

}

}