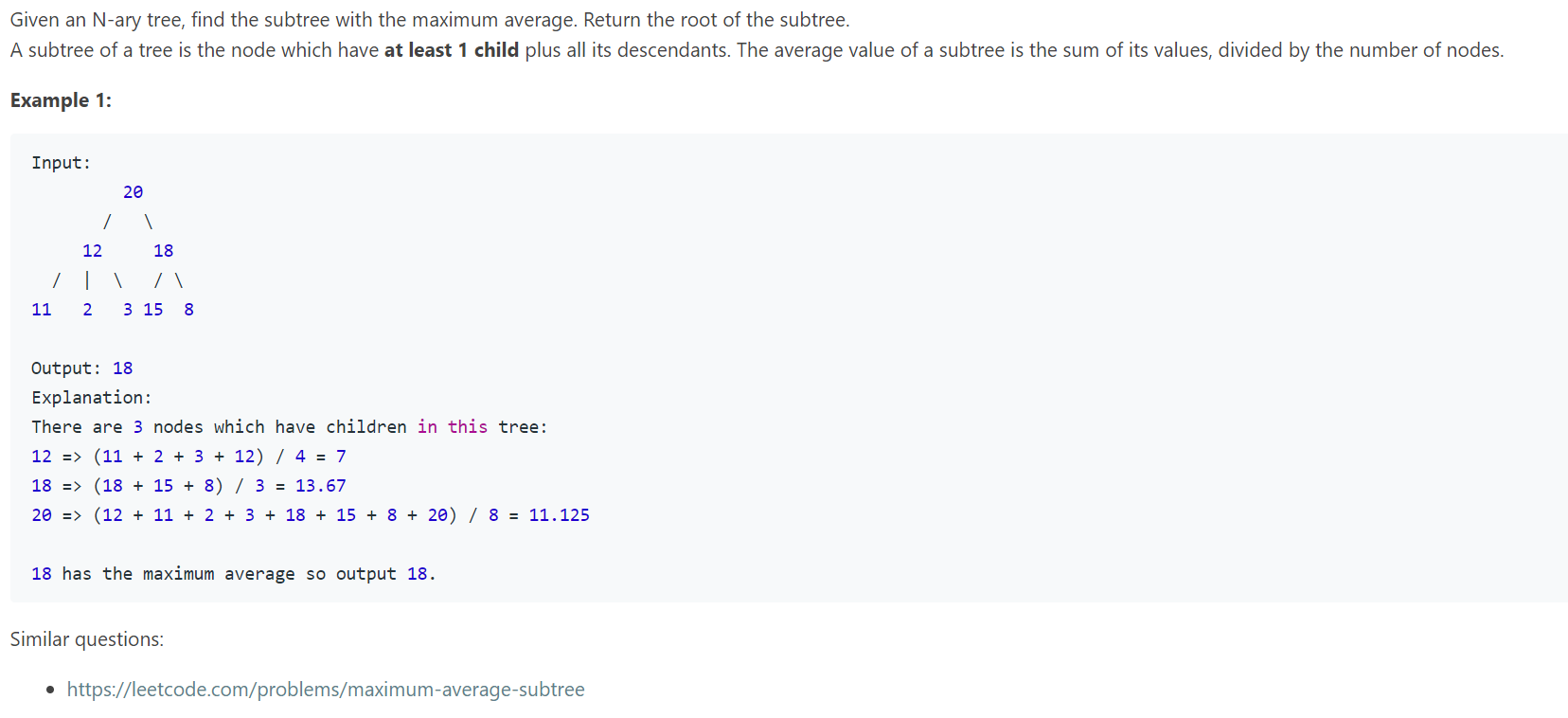
Amazon | OA 2019 | Subtree with Maximum Average

<https://leetcode.com/discuss/interview-question/349617>



SOL :

public class SubtreeWithMaximumAverage {

static class Node {

public int val;

public List<Node> children;

public Node() {}

public Node(int \_val) { val = \_val; }

public Node(int \_val, List<Node> \_children) {

val = \_val;

children = \_children;

}

}

double max;

Node maxNode;

public Node getMaximumAverage(Node root) {

maxNode = null;

max = Double.MIN\_VALUE;

helper(root);

return maxNode;

}

public double[] helper(Node root) {

if(root == null) return new double[]{0, 0};

double count = 1;

double sum = root.val;

if(root.children != null) {

for(Node child: root.children) {

double[] cur = helper(child);

sum += cur[0];

count += cur[1];

}

}

double average = sum / count;

if(count > 1 && average > max) {

max = average;

maxNode = root;

}

return new double[]{sum, count};

}

public static void main(String[] args) {

// Input:

// 20

// / \

// 12 18

// / | \ / \

// 11 2 3 15 8

Node left = new Node(12);

left.children = Arrays.asList(new Node(11), new Node(2), new Node(3));

Node right = new Node(18);

right.children = Arrays.asList(new Node(15), new Node(8));

Node root = new Node(20);

root.children = Arrays.asList(left, right);

test(root); // output: 18

}

private static void test(Node root) {

Node maxNode = new SubtreeWithMaximumAverage().getMaximumAverage(root);

System.out.println("Max Average: " + maxNode.val);

}

}