

```

import java.util.ArrayList;
class Node {
    int val;
    ArrayList<Node> children;
    public Node(int val){
        this.val = val;
        children = new ArrayList<Node>();
    }
}

public class MaximumSubtreeAverage {
    static class SumCount {
        int sum;
        int count;
        public SumCount(int s, int c) {
            sum = s;
            count = c;
        }
    }

    static double max = Double.MIN_VALUE;
    static Node res = null;
    public static Node find(Node node) {
        helper(node);
        return res;
    }

    public static SumCount helper(Node node) {
        if (node == null) return new SumCount(0, 0);
        if (root.children == null || root.children.size() == 0)
            return new SumCount(root.val, 1);
        int sum = node.val;
        int count = 1;
        for (Node n : node.children) {
            SumCount s = helper(n);
            sum += s.sum;
            count += s.count;
        }

        if (count > 1 && max < sum * 1.0 / (1.0 * count)) {
            max = sum * 1.0 / (1.0 * count);
            res = node;
        }
        return new SumCount(sum, count);
    }
}

```