链滴

给树形结构节点编号的一种思路

作者：flhuoshan

原文链接：https://ld246.com/article/1555657928503

来源网站：链滴

背景

- 假设需要遍历一棵深度为n的树形结构，给出各个节点层级，需要按照节点层级给每个节点编号。
- 真实的需求是：有一颗树形结构，深度优先遍历时，得到各节点的深度分别为：1,2,2,3,1,2,2,3,4,4,，此时给它们编号为

1，1.1，1.2，1.2.1，2，2.1，2.2，2.2.1，2.2.1.1，2.2.1.2，2.2.1.3。

主要思路

```
public static void main(String[] args) {

    //层级
    int[] levels = {1,2,2,3,1,2,2,3,4,4};
    int depth = 4;
    //计数器
    int[] counter = new int[depth];
    for(int i = 0; i < levels.length; i++){
        int curLevel = levels[i];
        //层级加一
        counter[curLevel - 1] += 1;

        //归零
        if(i > 0){
            for(int j = curLevel; j < counter.length; j++){
                counter[j] = 0;
            }
        }
        System.out.println(counter[0] + "." + counter[1] + "." + counter[2]+ "." + counter[3]);
    }
}
```

主要思路分析

数组levels代表要处理的数据，数组counter用于计数，counter[0]代表深度为1的计数器，counter[1代表深度为2的计数器，依次类推。该思路的平均时间复杂度为O(depth/2 * n)，是线性增长的。

改进思路

```
public static void main(String[] args) {

    //层级
    int[] levels = {1,2,2,3,1,2,2,3,4,4};
    int depth = 4;
    //计数器
    int[] counter = new int[depth];
    int lastLevel = 0;
    for(int i = 0; i < levels.length; i++){
        int curLevel = levels[i];
        //层级加一
        counter[curLevel - 1] += 1;
```

```
            //归零
            if(i > 0 && curLevel < lastLevel){
                for(int j = curLevel; j < counter.length; j++){
                    counter[j] = 0;
                }
            }
            lastLevel = curLevel;
            System.out.println(counter[0] + "." + counter[1] + "." + counter[2]+ "." + counter[3]);
        }
    }
```

改进思路分析

由于引入了lastLevel的概念，就可以据此判断是否必须归零。该思路的平均时间复杂度为O(depth/2 m * n),其中1<m<depth，它根据实际数据的不同而不同。