[LeetCode] 4 Keys Keyboard 四键的键盘

李博 bluemind   2017-12-06   1379浏览量

> **简介:**

Imagine you have a special keyboard with the following keys:

Key 1: (A): Print one 'A' on screen.

Key 2: (Ctrl-A): Select the whole screen.

Key 3: (Ctrl-C): Copy selection to buffer.

Key 4: (Ctrl-V): Print buffer on screen appending it after what has already been printed.

Now, you can only press the keyboard for N times (with the above four keys), find out the maximum numbers of 'A' you can print on screen.

Example 1:

```
Input: N = 3
Output: 3
Explanation:
We can at most get 3 A's on screen by pressing following key sequence:
A, A, A
```

Example 2:

```
Input: N = 7
Output: 9
Explanation:
We can at most get 9 A's on screen by pressing following key sequence:
A, A, A, Ctrl A, Ctrl C, Ctrl V, Ctrl V
```

Note:

1. 1 <= N <= 50
2. Answers will be in the range of 32-bit signed integer.

这道题给了我们四个操作，分别是打印A，全选，复制，粘贴。每个操作都算一个步骤，给了我们一个数字N，问我们N个操作最多能输出多个A。我们可以分析题目中的例子可以发现，N步最少都能打印N个A出来，因为我们可以每步都是打印A。那么能超过N的情况肯定就是使用了复制粘贴，这里由于全选和复制要占用两步，所以能增加A的个数的操作其实只有N-2步，那么我们如何确定打印几个A，剩下都是粘贴呢，其实是个trade off，A打印的太多

或太少，都不会得到最大结果，所以打印A和粘贴的次数要接近，<mark>最简单的方法就是遍历所有的情况然后取最大值</mark>，打印A的次数在[1, N-3]之间，粘贴的次数为N-2-i，加上打印出的部分，就是N-1-i了，参见代码如下：

解法一：

```
class Solution {
public:
    int maxA(int N) {
        int res = N;
        for (int i = 1; i < N - 2; ++i) {
            res = max(res, maxA(i) * (N - 1 - i));
        }
        return res;
    }
};
```

这道题也可以用DP来做，我们用一个一维数组dp，其中dp[i]表示步骤总数为i时，能打印出的最多A的个数，初始化为N+1个，然后我们来想递推公式怎么求。对于dp[i]来说，求法其实跟上面的方法一样，还是要遍历所有打印A的个数，然后乘以粘贴的次数加1，用来更新dp[i]，参见代码如下：

解法二：

```
class Solution {
public:
    int maxA(int N) {
        vector<int> dp(N + 1, 0);
        for (int i = 0; i <= N; ++i) {
            dp[i] = i;
            for (int j = 1; j < i - 2; ++j) {
                dp[i] = max(dp[i], dp[j] * (i - j - 1));
            }
        }
        return dp[N];
    }
};
```

这道题还有个O(1)时间复杂度的解法，好像利用了数学知识，不过博主貌似没太理解，参见这个帖子，哪位大神给博主讲解一下？

参考资料：

https://discuss.leetcode.com/topic/97764/o-1-time-o-1-space-c-solution-possibly-shortest-and-fastest

https://discuss.leetcode.com/topic/97628/java-4-lines-recursion-with-step-by-step-explanation-to-derive-dp

本文转自博客园Grandyang的博客，原文链接：[LeetCode] 4 Keys Keyboard 四键的键盘

，如需转载请自行联系原博主。