

## LeetCode 1374. 生成每种字符都是奇数个的字符串

### 1374. 生成每种字符都是奇数个的字符串

难度 简单

👍 2

❤

📄

🔖

🔔

💡

给你一个整数  $n$ ，请你返回一个含  $n$  个字符的字符串，其中每种字符在该字符串中都恰好出现 **奇数次**。

返回的字符串必须只含小写英文字母。如果存在多个满足题目要求的字符串，则返回其中任意一个即可。

示例 1:

输入:  $n = 4$

输出: "pppz"

解释: "pppz" 是一个满足题目要求的字符串，因为 'p' 出现 3 次，且 'z' 出现 1 次。当然，还有很多其他字符串也满足题目要求，比如: "ohhh" 和 "love"。

示例 2:

输入:  $n = 2$

输出: "xy"

解释: "xy" 是一个满足题目要求的字符串，因为 'x' 和 'y' 各出现 1 次。当然，还有很多其他字符串也满足题目要求，比如: "ag" 和 "ur"。

示例 3:

输入:  $n = 7$

输出: "holasss"

提示:

- $1 \leq n \leq 500$

```
/*  
n是奇数, 返回n个a  
n是偶数, 返回1个b, n-1个a  
*/  
class Solution {  
public:  
    string generateTheString(int n) {  
        string res;  
        if(n % 2 == 0) res += 'b', n --;  
        while(n --) res += 'a';  
        return res;  
    }  
};
```

## LeetCode 1375. 灯泡开关 III

---

### 1375. 灯泡开关 III

难度 中等

6



房间中有  $n$  枚灯泡，编号从 1 到  $n$ ，自左向右排成一排。最初，所有的灯都是关着的。

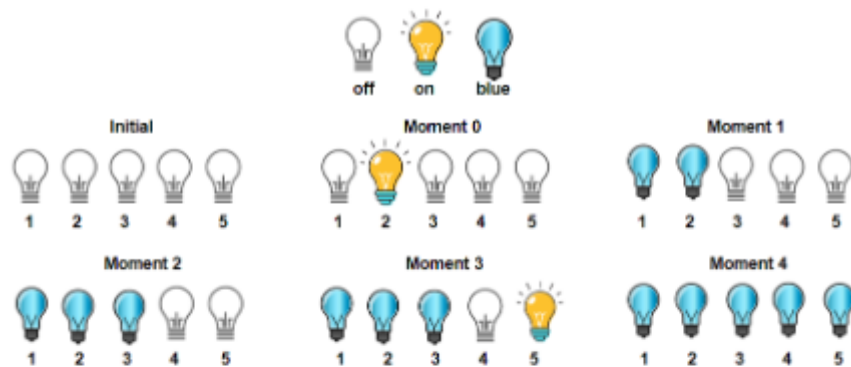
在  $k$  时刻 ( $k$  的取值范围是 0 到  $n - 1$ )，我们打开  $\text{light}[k]$  这个灯。

灯的颜色要想变成蓝色就必须同时满足下面两个条件：

- 灯处于打开状态。
- 排在它之前（左侧）的所有灯也都处于打开状态。

请返回能够让所有开着的灯都变成蓝色的时刻数目。

示例 1:



输入:  $\text{light} = [2, 1, 3, 5, 4]$

输出: 3

解释: 所有开着的灯都变蓝的时刻分别是 1, 2 和 4。

示例 2:

输入:  $\text{light} = [3, 2, 4, 1, 5]$

输出: 2

解释: 所有开着的灯都变蓝的时刻分别是 3 和 4 (index-0)。

```
/*
记录当前打开最大的值 == 扫描个数 就是满足要求了
最大值 = 下标 + 1
*/
class Solution {
public:
    int numTimesAllBlue(vector<int>& light) {
        int res = 0;
        int max = 0;
        for(int i = 0; i < light.size(); i++)
        {
            if(light[i] > max) max = light[i];
            if(max == i + 1) res++;
        }
    }
};
```

```
        return res;
    }
};
```

## LeetCode 1376. 通知所有员工所需的时间

### 1376. 通知所有员工所需的时间

难度 中等  13     

公司里有  $n$  名员工，每个员工的 ID 都是独一无二的，编号从 0 到  $n - 1$ 。公司的总负责人通过 `headID` 进行标识。

在 `manager` 数组中，每个员工都有一个直属负责人，其中 `manager[i]` 是第  $i$  名员工的直属负责人。对于总负责人，`manager[headID] = -1`。题目保证从属关系可以用树结构显示。

公司总负责人想要向公司所有员工通告一条紧急消息。他将会首先通知他的直属下属们，然后由这些下属通知他们的下属，直到所有的员工都得知这条紧急消息。

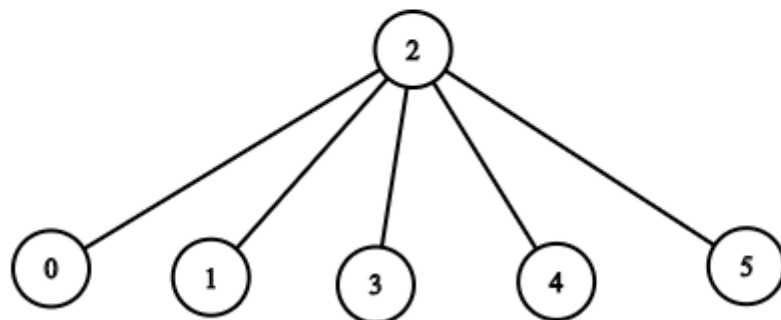
第  $i$  名员工需要 `informTime[i]` 分钟来通知它的所有直属下属（也就是说在 `informTime[i]` 分钟后，他的所有直属下属都可以开始传播这一消息）。

返回通知所有员工这一紧急消息所需要的分钟数。

#### 示例 1:

输入:  $n = 1$ , `headID = 0`, `manager = [-1]`, `informTime = [0]`  
输出: 0  
解释: 公司总负责人是该公司的唯一一名员工。

#### 示例 2:



```
/*
从根节点到叶节点的最长带权路径
两个vector存图
*/
class Solution {
public:
    vector<vector<int>> son;
```

```

    int numOfMinutes(int n, int headID, vector<int>& manager, vector<int>&
informTime) {
        son = vector<vector<int>>>(n);

        for(int i = 0; i < n; i++)
            if(i != headID)
                son[manager[i]].push_back(i);

        return dfs(headID, informTime);
    }

    int dfs(int u, vector<int>& informTime)
    {
        int res = 0;
        for(auto s : son[u]) res = max(res, dfs(s, informTime));
        return res + informTime[u];
    }
};

```