

## 题目：1419. 数青蛙

### 2. 讲解算法原理

解法：模拟

croak

crcoakroakcroak

crroak

↑

c	r	o	a	k
		1		



总结：

r, o, a, k → 找一下前驱字符，是否在哈希表中存在

存 → 前驱个数--，当前字符++  
不存 → 返回 -1

c → 找最后一个字符，是否在哈希表中存在

存 → 最后一个字符--，当前字符++  
不存 → 当前字符++

```
1 class Solution {
2 public:
3     int minNumberOfFrogs(string croakOfFrogs) {
4         // 0 -> c, 1 -> r, 2 -> o, 3 -> a, 4 -> k
5         int hash[5] = {0};
6         string str = "croak";
7         unordered_map<char, int> index;
8
9         // 初始化字符与索引的映射
10        for (int i = 0; i < 5; i++) {
11            index[str[i]] = i;
12        }
13        for (int i = 0; i < croakOfFrogs.size(); i++) {
14            char c = croakOfFrogs[i];
15            if (c == 'c') {
16                // 'c' 代表新的叫声开始
17                if (hash[4] != 0) hash[4]--; // 如果有完成的 "croak", 重新利用青蛙
18                hash[0]++;
19            } else {
20                // 检查其他字符是否按顺序出现
21                int charIndex = index[c];
22                if (hash[charIndex - 1] > 0) {
23                    hash[charIndex - 1]--;
24                    hash[charIndex]++;
25                } else {
26                    // 前一个字符不足, 说明顺序有问题
27                    return -1;
28                }
29            }
30        }
31
32        // 检查所有未完成的叫声
33        for (int i = 0; i < 4; i++) {
34            if (hash[i] != 0) return -1;
35        }
36
37        return hash[4]; // 返回最大并发青蛙数量
38    }
```

