LeetCode(91. Decode Ways) 题解

BitBrave

2019 年 6 月 22

Question 1

Question

给出一段文字由 $A-Z$ 组成，分别使用 1-26 去编码。现在给出这样一段编码，去找出有多少种方式去解码它，即有多少种字母的组合方式可以被编码成给出的样本的序列。

Solution

总共有 26 个字母，则数组有 $1, 2, 3, ..., 26$。我们设定给出的编码为 A，长度为 n，$OPT(A, i)$ 表示 A 中前 i 个字符所能形成字母组合方式数量，则我们要求的是 $OPT(A, n)$。我们建立算法 *Decoding*，描述如下：

1. 设置 $OPT(A, 0) = 1$，$OPT(A, 1) = 1$。若 $n > 1$，执行步骤 2。
2. 计算 $OPT(A, i)$，其表达式为————

$$OPT(A,i) = \begin{cases} OPT(A,i-1) & if\ A_{i-1} \geq 3\ or\ A_{i-1} == 0\ or\ (A_{i-1} == 2\ and\ A_i \geq 7) \\ OPT(A,i-1) + OPT(A,i-2) & if\ A_{i-1} == 1\ or\ (A_{i-1} == 2\ and\ A_i \leq 6) \end{cases}$$

具体算法如下。

Algorithm

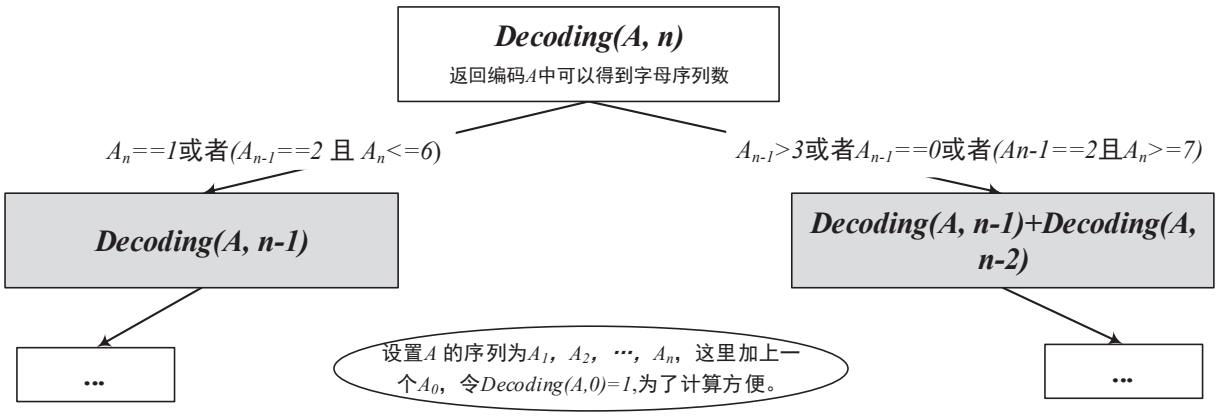
输入： A：编码序列，n：长度
输出： 字母组合种类数

```
1: function DECODING(A, n)
2:     OPT(A, 0) ← 1
3:     if n == 1 then
4:         OPT(A, 1) ← 1
5:         return OPT(A, 1)
6:     end if
7:     OPT(A, 1) ← 1
```

```
    for i=2 to n do
        if A_{i-1} ≥ 3 or A_{i-1} == 0 or (A_{i-1} == 2 and A_i ≥ 7) then
            OPT(A, i) ← OPT(A, i − 1)
            return OPT(A, 1)
        end if
        if A_{i-1} == 1 or (A_{i-1} == 2 and A_i ≤ 6) then
            OPT(A, i) ← OPT(A, i − 1) + OPT(A, i − 2)
        end if
    end for
    return OPT(A, n)
end function
```

Subproblem reduction graph

分解图如下————



Prove the correctness

算法 *Decoding(A,n)* 通过求解子问题的方式获得自身的解，那么我们只要证明问题与子问题之间的转移关系是正确的，再获得最基本的子问题的解，即可说我们的求解方式是正确的。假设现在 $OPT(A,i)$ 表示 A 中前 i 个数字所代表的字母编码序列数。如果现在再在后面接一个数字，那我们就需要求 $OPT(A,i+1)$，分析如下————

1. 如果 $A_i \geq 3$ *or* $A_i == 0$ *or* $(A_i == 2$ *and* $A_i \geq 7)$，那么后面接上的数字不能与前面的数字组成 $1-26$ 的数字，也就不能形成组合成有效的字母编码。所以加上之后原有的编码种类并没有变化，只是末尾多了一个字母编码而已。此时，$OPT(A,i+1) = OPT(A,i)$。

2. 如果 $A_i == 1$ *or* $(A_i == 2$ *and* $A_{i+1} \leq 6$，那么后面接上的数字可以与前面的数字组成 $1-26$ 的数字，也就是说可以组合成有效的字母编码。那现在就存在两种情况，一是加上的数字单独是一个字母的编码这时，总的编码种类数还是 $OPT(A,i)$，而如果将 A_i 和 A_{i+1} 合起来看做一个字符的编码，则此时的种类数应该是 $OPT(A,i-1)$。所以这种情况下，加入一个数字之后，$OPT(A,i+1) = OPT(A,i) + OPT(A,i-1)$。

Analyse the complexity

算法 *Decoding(A,n)* 类似于斐波那契数列，我们用数组记录下来了每个子问题的解，因此只用一次 for 循环就得到了结果。因此时间复杂度为——

$$T(n) = O(n)$$

代码如下

Runtime: 4 ms, faster than 85.7% of C++ online submissions for Decode Ways. Memory Usage: 8.6 MB, less than 27.02% of C++ online submissions for Decode Ways.

```
class Solution {
public:
int numDecodings(string s) {
int len = s.size();
if(len == 0 || s[0] == '0') return 0;
vector<int> res(len+1, 1);
for(int i=1; i<len; i++){
if((s[i-1]=='0' || s[i-1]>='3') && s[i]=='0') return 0;
if((s[i-1]=='1' && s[i]!='0')||(s[i-1]=='2' && s[i]<='6' && s[i]!='0')) res[i+1] = res[i] + res[i-1];
else if(s[i]=='0' && (s[i-1]=='2'||s[i-1]=='1')) res[i+1] = res[i-1];
else res[i+1] = res[i];
}
return res[len];
}
};
```