

# 91. Decode Ways

题目描述: <https://leetcode.com/problems/decode-ways/>

编码把

```
A -> 1
B -> 2
...
Z -> 26
```

给定一个编码之后的序列，求解码种数

解题思路:

$f[i]$ 代表截止到 $i$ 位置为止，共有多少种翻译方式。

注意会有0的存在，因为10和20都是合法的

如果  $i$  位置自己本身合法，且和前一个组成的两位数合法，则  $f[i] = f[i-1] + f[i-2]$

如果  $i$  位置自己本身合法，但和前一个组成的两位数不合法，则  $f[i] = f[i-1]$

如果  $i$  位置自己本身不合法，但和前一个组成的两位数合法，则  $f[i] = f[i-2]$

如果  $i$  位置自己本身不合法，且和前一个组成的两位数不合法，则证明本串不可翻译 return 0

代码:

```
class Solution {
public:
    int numDecodings(string s) {
        vector<int> a(100, 0);
        for(int i = 1; i <= 26; i++){
            a[i] = 1;
        }
        vector<int> f(s.size() + 1, 0);
        if(s.size() == 0)
            return 0;
        char c;
        char bc = s.at(0);
        f[1] = a[bc - '0'];
        f[0] = 1;
        for(int i = 1; i < s.size(); i++){
            c = s.at(i);
            bc = s.at(i-1) + s.at(i);
```

```

        dc = s.at(i-1);
        int ic = c - '0';
        int ibc = bc - '0';
        int combine = ibc * 10 + ic;
        if(a[ic] > 0 && ibc > 0 && a[combine] > 0)
            f[i+1] = f[i-1]+f[i];
        else if(a[ic] > 0 && (ibc <= 0 || a[combine] == 0))
            f[i+1] = f[i];
        else if(a[ic] <= 0 && ibc > 0 && a[combine] > 0)
            f[i+1] = f[i-1];
        else
            return 0;
    }
    return f[s.size()];
}
};

```