

```
class Solution {
```

```
    public int numDecodings(String s) {
```

```
        if (s.charAt(0) == '0') return 0; // 开头为0，返回0？
```

```
        int[] dp = new int[s.length() + 1];
```

```
        dp[0] = dp[1] = 1; // 不为0的字目，第一个可以有一种解法
```

```
        for (int i = 2; i <= s.length(); i++) {
```

```
            // 如果该位不为'0'，说明该位单独成字母合法
```

```
            if (s.charAt(i - 1) != '0') {
```

```
                dp[i] += dp[i - 1];
```

```
            }
```

```
            // 如果后两位能组成"1x" (x为任意数字) 或者 "2x" (x小于7)，说明最后两位组成字母合法
```

```
            if ((s.charAt(i - 2) == '1') || (s.charAt(i - 2) == '2' && s.charAt(i - 1) <= '6')) {
```

```
                dp[i] += dp[i - 2];
```

```
            }
```

```
        }
        return dp[s.length()];
```

```
    }
```

```
}
```

判断单个数字能

否解码，

对应:  $dp[i-1]$

判断两个数字能

否解码，

对应:  $dp[i-2]$

2 2 2 6  $\Rightarrow$  2 2  $i=2$   $dp[i] += dp[i-1] = 1$

$\downarrow$

2 + 22  
1种 1种

$\Rightarrow$  2种  $\Rightarrow dp[i] += dp[i-1]$

$i=2$

$dp[i] += dp[i-1]$

$dp[i] += dp[i-2]$

$\Rightarrow$  2 2 2

$\downarrow$

2 2 2  
2种 1种

$\Rightarrow$  3种

$dp[i] += dp[i-1]$

$i=3$

$dp[i] += dp[i-1]$

$dp[i] += dp[i-2]$

$\Rightarrow$  2 2 2 6

2 2 + 2 6 or 2 2 2 + 6  $\Rightarrow$  5种

2种

3种

$dp[i] += dp[i-1]$

$i=4$

$dp[i] += dp[i-1]$

$dp[i] += dp[i-2]$