# 统计一

## 问题描述

NewCode总是力争上游，凡事都要拿第一，所以他对“1”这个数情有独钟。爱屋及乌，他也很喜欢包含1的数，例如10、11、12、……。

例如：N=2，1、2出现了1个"1"。N=12，1、2、3、4、5、6、7、8、9、10、11、12。出现了5个"1"。你能帮他统计一下整数里有多少个1吗？

## 解题思路

### 解法一：

最直接的方法就是从1开始遍历到N，将其中每一个数中含有"1"的个数加起来，就得到了问题的解。

|  |
| --- |
| **private static long** countOne3(**long** n) {  **long** count = **0**;  **for** (**int** i = **0**, j; i <= n; i++) {  j = i;  **while** (j != **0**) {  **if** (j % **10** == **1**) {  count++;  }  j = j / **10**;  }  }  **return** count; } |

此方法简单，容易理解，但它的问题是效率，时间复杂度为，N比较大的时候，需要耗费很长的时间。

### 解法二：

我们重新分析下这个问题，对于任意一个个位数n，只要n≥1,它就包含一个"1"；n<1，即n=0时，则包含的"1"的个数为0。于是我们考虑用分治的思想将任意一个n位数不断缩小规模分解成许多个个位数，这样求解就很方便。

但是，我们该如何降低规模？仔细分析，我们会发现，任意一个n位数中"1"的个位可以分解为两个n-1位数中"1"的个数的和加上一个与最高位数相关的常数C。例如，f(12) = f(10 - 1) + f(12 - 10) + 3,其中3是表示最高位为1的数字个数，这里就是10、11、12；f(132)=f(100 -1) + f(132 - 100) + 33，33代表最高位为1的数字的个数，这里就是100~132；f(232) = 2\*f(100 - 1) + f(32) + 100，因为232大于199，所以它包括了所有最高位为1的数字即100~199，共100个。

综上，我们分析得出，最后加的常数C只跟最高位n1是否为1有关，当最高位为1时，常数C为原数字N去掉最高位后剩下的数字+1，当最高位为1时，常数C为10bit，其中bit为N的位数-1，如N=12时，bit=1，N=232时，bit=2。

于是，我们可以列出递归方程如下：

递归的出口条件为：

http://www.nowamagic.net/librarys/veda/detail/1064