# 题目

编写一个方法，计算从 0 到 n (含 n) 中数字 2 出现的次数。

示例:

输入: 25

输出: 9

解释: (2, 12, 20, 21, 22, 23, 24, 25)(注意 22 应该算作两次)

提示：

n <= 10^9

# 分析

## 方法一：数学

思路：

要计算从0到n（包括n）中数字2出现的次数，可以通过分析每个位置上数字2出现的规律来解决这个问题。以下是解决方法的思路：

1、对于每个位置i（从右往左数，从0开始），将n分为两部分，高位部分high和低位部分low，以i为分界线，即n = high \* 10^i + low。

2、分三种情况讨论：

- 当i位置上的数字小于2时，出现2的次数仅由高位high决定，计算公式为high \* 10^i。

- 当i位置上的数字等于2时，出现2的次数由高位high和低位low共同决定，计算公式为high \* 10^i + low + 1。

- 当i位置上的数字大于2时，出现2的次数仅由高位high决定，计算公式为(high + 1) \* 10^i。

3、最后累加每个位置上2出现的次数即可得到结果。

下面是实现该算法的代码：

class Solution {

public:

int numberOf2sInRange(int n) {

int count = 0;

long long factor = 1; // 用于表示当前位的权值

while (n / factor > 0) {

int high = n / (factor \* 10); // 高位

int cur = (n / factor) % 10; // 当前位

int low = n % factor; // 低位

if (cur < 2) {

count += high \* factor;

} else if (cur == 2) {

count += high \* factor + low + 1;

} else {

count += (high + 1) \* factor;

}

factor \*= 10;

}

return count;

}

};

这段代码使用了一个factor变量来表示当前位置的权值，通过不断除以10来移动到下一个位置。在每个位置上，根据当前位的值和高低位的情况来计算2出现的次数，并累加到count中。最后返回count即可。