题目：

Given an integer n, count the total number of digit 1 appearing in all non-negative integers less than or equal to n.

For example:

Given n = 13,

Return 6, because digit 1 occurred in the following numbers: 1, 10, 11, 12, 13.

[思路]

reference: https://leetcode.com/discuss/44281/4-lines-o-log-n-c-java-**Python**

intuitive: 每 10 个数, 有一个个位是 1, 每 100 个数, 有 10 个十位是 1, 每 1000 个数, 有 100 个百位是 1. 做一个循环, 每次计算单个位上 1 得总个数(个位,十位, 百位).

例子:

以算百位上 1 为例子: 假设百位上是 0, 1, 和 >=2 三种情况:

case 1: n=3141092, a= 31410, b=92. 计算百位上 1 的个数应该为 3141 *100 次.

case 2: n=3141192, a= 31411, b=92. 计算百位上 1 的个数应该为 3141 *100 + (92+1) 次.

case 3: n=3141592, a= 31415, b=92. 计算百位上 1 的个数应该为 (3141+1) *100 次.

以上三种情况可以用 一个公式概括:

(a + 8) / 10 * m + (a % 10 == 1) * (b + 1);

```
class Solution {
    typedef long long int64_t;
public:
    int countDigitOne(int n) {
        int result = 0;
        for (int64_t m = 1; m <= n; m *= 10){
            int a = n / m, b = n % m;
            result += (a + 8) / 10 * m;
            if (a % 10 == 1) result += (b + 1);
        }
        return result;
    }
};
```