

有趣的数字和 ☆☆

题目描述

如果一个正整数的二进制表示包含奇数个 1，那么小A就会认为这个正整数是有趣的。

例如，7 的二进制表示为 $(111)_2$ ，包含 1 的个数为 3 个，所以 7 是有趣的。但是 $9=(1001)_2$ 包含 2 个 1，所以 9 不是有趣的。

给定正整数 l, r ，请你统计满足 $l \leq n \leq r$ 的有趣的整数 n 之和。

输入输出格式

输入：一行，两个正整数 l, r ，表示给定的正整数。

输出：一行，一个正整数，表示 l, r 之间有趣的整数之和。

输入示例	输出示例
3 8	19
5 15	70
1 10	25

样例解释

十进制	二进制	1的个数	是否有趣
3	0011	2个	否
4	0100	1个	是
5	0101	2个	否
6	0110	2个	否
7	0111	3个	是
8	1000	1个	是

在区间 $[3, 8]$ 中，有趣的数字有：4, 7, 8，它们的和为 $4 + 7 + 8 = 19$ 。

算法分析

1. 暴力解法

最直接的方法是遍历区间 $[l, r]$ 中的每个数，检查其二进制表示中 1 的个数是否为奇数，如果是则累加。

```
long long sum = 0;
for (int i = l; i <= r; i++) {
    if (check(i)) { // check函数用于判断i是否有趣
        sum += i;
    }
}
```

这种方法的时间复杂度为 $O((r - l + 1) \cdot \log r)$ ，当 $r = 10^9, l = 1$ 时运算量约为 10^9 级别，会导致超时。

2. 数学规律解法

- (1) **问题拆解:** 将计算区间 $[l, r]$ 的问题转化为计算两个前缀和的差: $f(r) - f(l-1)$, 其中 $f(x)$ 表示从 0 到 x 的所有有趣数之和。所以用 $\text{res}(l, r)$ 表示从 l 到 r 的有趣数字和, 则有:

$$\text{res}(l, r) = f(r) - f(l-1)$$

- (2) **分组讨论:** 观察二进制数的规律, 可以发现每 4 个连续整数中, 有趣数的分布具有周期性:

十进制	二进制	1的个数	是否有趣
0	0000	0个	否
1	0001	1个	是
2	0010	1个	是
3	0011	2个	否

十进制	二进制	1的个数	是否有趣
4	0100	1个	是
5	0101	2个	否
6	0110	2个	否
7	0111	3个	是

十进制	二进制	1的个数	是否有趣
8	1000	1个	是
9	1001	2个	否
10	1010	2个	否
11	1011	3个	是

十进制	二进制	1的个数	是否有趣
12	1100	2个	否
13	1101	3个	是
14	1110	3个	是
15	1111	4个	否

观察上述四组数据, 可以发现:

- 对于任意四位连续的数字: ...00, ...01, ...10, ...11, 其中省略高位部分
- 如果省略的部分中有奇数个1, 那么有趣的数字是: ...01 和 ...10
- 如果省略的部分中有偶数个1, 那么有趣的数字是: ...00 和 ...11
- 因为 $\dots00 + \dots11 = \dots01 + \dots10$
- 所以无论省略的部分中是奇数个1 还是偶数个1, 每组四个数中有趣数字之和都等于四个数组之和的一半, 也就是: $\frac{\dots00 + \dots01 + \dots10 + \dots11}{2}$
- 所以

$$\text{res}[0, 3] = \frac{0+1+2+3}{2} = 3$$

$$\text{res}[0, 7] = \frac{0+1+2+3+4+5+6+7}{2} = 14$$

$$\text{res}[0, 11] = \frac{0+1+2+\dots+11}{2} = 33$$

$$\text{res}[0, x] = \frac{0+1+2+\dots+x}{2} = \frac{x*(x+1)}{4}$$

($x \bmod 4 == 3$ 确保正好按照4个4个一组分完所有数字)

- (3) **求和关系:** 对于可以完整分成的 4 个数一组, 该组中有趣数字之和等于所有数字之和的一半。

```
res += (x + 1) * x / 4; // 核心公式: 完整组的和
```

(4) **边界处理:** 对于无法组成完整 4 个一组的剩余部分, 需要单独判断每个数是否有趣并累加。

```
bool check(int x) { // 判断x是否有趣
    int cnt = 0;
    while (x) {
        if (x & 1) cnt++;
        x >= 1;
    }
    return (cnt & 1);
}

while (x > 0 && x % 4 != 3) {
    //从大开始缩小数字, 直到这个数字可以完整按照4个4个分
    if (check(x)) res += x--; // 处理边界数字
}
```

代码运行步骤展示

以输入 5 17 为例: $res(5, 17) = f(17) - f(4)$

1. 计算 $f(17)$:

- $17 \bmod 4 = 1 \neq 3$, 需要边界处理
- 从17开始向前处理不完整组:
 - 17: 二进制(10001), 有2个1, 偶数个, 不是有趣的
 - 16: 二进制(10000), 有1个1, 奇数个, 是有趣的, $res += 16$.
 - 15: $15 \bmod 4 = 3$, 停止边界处理.
- 此时 $x = 15$, $15 \bmod 4 = 3$, 可以使用公式($0-3, 4-7, 8-11, 12-15$ 都是完整组)
- 完整组的和 = $\frac{15*16}{4} = 60$
- 所以 $f(17) = 16 + 60 = 76$

2. 计算 $f(4)$:

- $4 \bmod 4 = 0 \neq 3$, 需要边界处理
- 从4开始向前处理不完整组:
 - 4: 二进制(100), 有1个1, 奇数个, 是有趣的, $res += 4$.
 - 3: $3 \bmod 4 = 3$, 停止边界处理.
- 此时 $x = 3$, $3 \bmod 4 = 3$, 可以使用公式($0-3$ 是完整组)
- 完整组的和 = $\frac{3*4}{4} = 3$
- 所以 $f(4) = 4 + 3 = 7$

3. 最终结果: $res(5, 17) = f(17) - f(4) = 76 - 7 = 69$

拓展思考

- 除了寻找数学规律, 这种按位的题目还可以考虑使用数位dp来解决。