# 题目

递归乘法。写一个递归函数，不使用 \*运算符， 实现两个正整数的相乘。可以使用加号、减号、位移，但要吝啬一些。

示例1:

输入：A = 1, B = 10

输出：10

示例2:

输入：A = 3, B = 4

输出：12

提示:

保证乘法范围不会溢出

# 分析

## 方法一：递归

思路：

这个递归函数的思路是，如果B为0，则返回0；如果B为负数，则返回-A和-B的乘积的相反数；如果B为偶数，则返回A左移一位（相当于A乘以2）和B右移一位（相当于B除以2）的乘积；如果B为奇数，则返回A加上A左移一位和B右移一位的乘积。

代码：

class Solution {

public:

int multiply(int A, int B) {

if (B == 0) {

return 0;

} else if (B < 0) {

return -multiply(A, -B);

} else if (B % 2 == 0) {

return multiply(A << 1, B >> 1);

} else {

return A + multiply(A << 1, B >> 1);

}

}

};