

Java方向编程题答案

####

day15

[编程题]25083-另类加法

链接: <https://www.nowcoder.com/questionTerminal/e7e0d226f1e84ba7ab8b28efc6e1aebc>

【题目解析】

实现两个数的相加求和，实现过程中不能用+运算符

【解题思路】

二进制加法特点。

- 位的异或运算跟求'和'的结果一致：

异或 $1 \wedge 1 = 0$ $1 \wedge 0 = 1$ $0 \wedge 0 = 0$
求和 $1 + 1 = 0$ $1 + 0 = 1$ $0 + 0 = 0$

- 位的与运算跟求'进位'的结果一致：

位与 $1 \& 1 = 1$ $1 \& 0 = 0$ $0 \& 0 = 0$
进位 $1 + 1 = 1$ $1 + 0 = 0$ $0 + 0 = 0$

【示例代码】

```
import java.util.*;

public class UnusualAdd {
    public int addAB(int A, int B) {
        if(B==0)
            return A;
        int sum = A ^ B; //相加但不进位
        int jin = (A & B) << 1; //进位但不相加
        return addAB(sum, jin);
    }
}
```

[编程题]45843-饥饿的小易

链接: <https://www.nowcoder.com/questionTerminal/5ee8df898312465a95553d82ad8898c3>

【题目解析】

无

【解题思路】

小易的移动公式:

$$f(x) = 4 * x + 4$$
$$g(x) = 8 * x + 7$$

计算可以得出两个规律：

1. $g(f(x)) = f(g(x))$ 即f和g的执行顺序没有影响
2. $f(f(f(x))) = g(g(x))$ 即做3次f的变换等价于做2次g的变换

由规律1可以得出对于一个可行方案，可以调整其变换顺序。比如：ffggfggff 可以转换为fffffgggg

由规律2并且为了减少执行次数，每3个f可以转换为2个g 如方案fffffgggg可以转换为ffggggggg.

因此一个最优的策略：f的执行次数为 0, 1, 2。对于输入x，只需要要求x, 4x+3, 4(4x+3)+3 的最小g执行次数即可。

[示例代码]

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        long x0 = scanner.nextLong();
        long m = 1000000007; //取模的值
        long s = 100000; //神秘力量使用的次数

        long[] begin = new long[3]; //f(x) = 4x+3 执行3次

        //3次的取值
        begin[0] = x0;
        begin[1] = (4 * begin[0] + 3) % m;
        begin[2] = (4 * begin[1] + 3) % m;

        long minStep = s;
        long cur = 0;
        int step = 0; //执行的步数
        for (int i = 0; i < 3; i++) {
            cur = begin[i];
            step = i;
            while (cur != 0 && step < minStep) {
                cur = (8 * cur + 7) % m; //g(x) = 8x+7 执行
                step++;
            }
            minStep = minStep < step ? minStep : step;
        }
        if (minStep < s) { //如果执行步长没有超过s输出最小步长
            System.out.println(minStep);
        } else { //超过返回-1
            System.out.println(-1);
        }
    }
}
```

比特科技制作