博弈论新总结

公平组合游戏

需要满足的三个条件：

1.由两名玩家交替行动

2.在游戏进程的任意时刻，可以执行的合法行动与轮到哪名玩家无关

3.游戏中的同一个状态不可能多次抵达，游戏以玩家无法行动为结束，且游戏一定会在优先步后以非平局结束。

博弈图

一个有向无环图，图中有一个唯一的起点，每次操作相当于沿着有向边移动，无法移动判负，则称为有向图游戏。

任意一个公平组合游戏都可以转化为有向图游戏，每个局面看成一个点。

当绘制出博弈图后，可以用 $O(n+m)$ 的时间得到每个状态是必胜还是必败□ n 为状态种数□ m 为边数。

$Beatty$ 定理

如果两个无理数 a,b 满足□ ${\frac 1 a}+{\frac 1 b}=1$ □那么对于两个集合 A,B □ $A={\lfloor {na} \rfloor},B={\lfloor {nb} \rfloor},n∈Z$ □有 $A∩B=∅,A∪B=N^{+}$

例1

威佐夫博弈的高精度模板题，到 10^{100} 级别。

我们知道威佐夫博弈的必败态当且仅当 ${\lfloor {\frac {(b-a)×({\sqrt {5}}+1)} {2}} \rfloor}==a$ □直接上 $Java$ □二分求一下 ${\sqrt {5}}$ □剩下暴力计算。

```
import java.math.*;
import java.util.*;

public class Main {
    public static void main(String[] args) {
        BigDecimal one = BigDecimal.valueOf(1);
        BigDecimal two = BigDecimal.valueOf(2), five =
BigDecimal.valueOf(5);
        BigDecimal t = one.add(sqrt(five, 500)).divide(two);
        Scanner sc = new Scanner(System.in);
```

```
        while (sc.hasNext()) {
            BigDecimal a, b, tmp = null;
            a = sc.nextBigDecimal();
            b = sc.nextBigDecimal();

            if (a.compareTo(b) > 0) {
                tmp = a;
                a = b;
                b = tmp;
            }

            if (b.subtract(a).multiply(t).setScale(0,
BigDecimal.ROUND_DOWN).equals(a)) {
                System.out.println(0);
            } else
                System.out.println(1);

        }

        sc.close();
    }

    private static BigDecimal sqrt(BigDecimal x, int n) {
        BigDecimal l = BigDecimal.ZERO, r = x, mid;
        BigDecimal two = BigDecimal.valueOf(2);

        for (int i = 0; i <= n; i++) {
            mid = l.add(r).divide(two);

            if (mid.pow(2).compareTo(x) <= 0)
                l = mid;
            else
                r = mid;
        }

        return l;
    }

}
```