斐波那契博弈

- 題型
  - 有N個物品，二人輪流取物，先手最少拿一最多無上限，但不能取完
  - 之後每次取物品數不能超過上次的2倍且至少一個
  - 取走最後一件者勝利
- 結論
  - N 為斐波那契數時後手必勝
  - 非斐波那契數時後手必敗

巴什博奕 (Bash Game)

- 題型
  - 有N個物品，二人輪流取物，每次拿A~B個，少於A必須全拿
  - 取走最後一件者勝利
- 結論
  - $0 \leq N\%(A+B) < A$ 後手勝利
- 詳解
  - 如果 $A < N \leq A+B$ 後手必勝
  - $N = (A+B) \times X + Y$ 當$Y = 0$時 後手必勝
  - 所以 $0 \leq Y < A$ 後手勝利

威佐夫博弈 (Wythoff's Game)

- 題型
  - 有二堆物品初始值(X,Y)，二人輪流從其中一堆取至少一件，至多不限
  - 或從二堆中取相同件物品
  - 最後取完者勝
- 結論
  - $Z = |Y - X|$
  - $W = \frac{\sqrt{5}+1}{2} \times Z$
  - $W == min(X, Y)$ 後手必勝
- 詳解

尼姆博弈 (Nim Game)

- 題型
  - 有任意堆物品，每堆物品個數是任意的
  - 雙方輪流取物，每次從一堆中取部分或全部物品，最少一件
  - 取道最後一件者勝利
- 結論
  - 把每堆物品數xor起來，數字為0則後手必勝
- 詳解

SG定理

- mex運算
  - 找出不屬於當前集合最小的非負整數
  - Ex: $mex1, 2, 3 = 0$, $mex0, 1, 3 = 2$
- 元素X的後繼狀態表示為S
  - S表示X可以轉移過去的狀態
- SG函式
  - 對於任意狀態X，定義函式$SG(X) = mex(S)$
- SG定理
  - 遊戲和的SG函式等於各個遊戲SG函式的Nim和
- SG函式和博弈結果的關係
  - 在一個遊戲中有N個物品，如果$SG(N) = 0$ 後手必勝
  - K個遊戲組成的一個遊戲，遊戲和的SG函式(各個遊戲SG函式的Nim和)等於0 後手必勝

Ex

- 題目
  - 有三堆石頭，數量為 m, n, p
  - 二人輪流選擇任意一堆拿走F個
  - F為斐波那契數列中的玩素
  - 先取光者勝
- 解析

    - 看成三個小遊戲，使用SG定理

- Code

```
int F[N+10],SG[MAXN+10],S[MAXN+10];
void getSG(int K){
    memset(SG,0,sizeof(SG));
    for(int i = 1 ; i <= K ; i++ ){
        memset(S,0,sizeof(S));
        // S為狀態i的後繼狀態
        for(int j = 0 ; F[j]<= i && j < N ; j++ )
            S[ SG[i-F[j]] ] = 1;
        // SG(i) = mex(S)
        for(int j = 0 ;; j++ ) if( !S[j] ){
            SG[i] = j;
            break;
        }
    }
}

int main(){

    F[N]; // Fibonacci 打表
    getSG(MAXN);

    int n,m,p;
    while( ~scanf("%d %d %d",&m,&n,&p) ){
        // 將三個遊戲組合在一起
        if( SG[m]^SG[n]^SG[p] )
            print("First move Win");
        else
            print("Back hand Win");
    }

}
```