ACM程序设计

杭州电子科技大学 刘春英

acm@hdu.edu.cn

今天，

你 AC 了吗？

每周一星（10）：

<u>Lin2144</u>

第十一讲

组合博弈入门

（Simple Game Theory）

导引游戏

(1) 玩家：2人；
(2) 道具：23张扑克牌；
(3) 规则：

- 游戏双方轮流取牌；
- 每人每次仅限于取1张、2张或3张牌；
- 扑克牌取光，则游戏结束；
- 最后取牌的一方为胜者。

基本思路？

请陈述自己的观点

第一部分

简单取子游戏

（组合游戏的一种）

什么是组合游戏——

(1) 有两个玩家；
(2) 游戏的操作状态是一个有限的集合（比如：限定大小的棋盘）；
(3) 游戏双方轮流操作；
(4) 双方的每次操作必须符合游戏规定；
(5) 当一方不能将游戏继续进行的时候，游戏结束，同时，对方为获胜方；
(6) 无论如何操作，游戏总能在有限次操作后结束；

概念:必败点和必胜点(P点 & N点)

- **必败点(P点) :**前一个选手(**P**revious player)将取胜的位置称为必败点。

- **必胜点(N点) :下**一个选手(Next player)将取胜的位置称为必胜点。

必败(必胜)点属性

(1) 所有终结点是必败点（P点）；

(2) 从任何必胜点（N点）操作，至少有一种方法可以进入必败点（P点）；

(3)无论如何操作， 从必败点（P点）都只能进入必胜点（N点）.

取子游戏算法实现——

步骤1:将所有终结位置标记为必败点（P点）；

步骤2: 将所有一步操作能进入必败点（P点）的位置标记为必胜点（N点）

步骤3:如果从某个点开始的所有一步操作都只能进入必胜点（N点），则将该点标记为必败点（P点）；

步骤4: 如果在步骤3未能找到新的必败（P点），则算法终止；否则，返回到步骤2。

课内练习:

- Subtraction Games:
  subtraction set S = {1, 3, 4}

x :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14...
Pos:	P	N	P	N	N	N	N	P	N	P	N	N	N	N	P...

实战练习…

- **kiki's game**

第二部分

Nim游戏

Nim游戏简介

(1) 有两个玩家；

(2) 有三堆扑克牌（比如：可以分别是5，7，9张）；

(3) 游戏双方轮流操作；

(4) 玩家的每次操作是选择其中某一堆牌，然后从中取走任意张；

(5) 最后一次取牌的一方为获胜方；


Winnie The Pooh
Pooh & Piglet

初步分析

- (0, 0, 0)
- (0, 0, x)
- (0, 1, 1)
- (0, k, k)
- (14, 35, 46)

- P-position
- N-position
- P-position
- P-position
- ???

引入概念：Nim-Sum

- **定义: 假设** $(x_m \cdots x0)_2$ **和**$(y_m \cdots y_0)_2$ **的**nim-sum**是**$(z_m \cdots z_0)_2$,**则我们表示成** $(x_m \cdots x_0)_2 \oplus (y_m \cdots y_0)_2 = (z_m \cdots z_0)_2$, **这里**，$z_k = x_k + y_k \pmod 2$ (k=0…m) .

$$
\begin{array}{r}
22 = 10110_2 \\
51 = 110011_2 \\
\hline
\text{nim-sum} = 100101_2 = 37
\end{array}
$$

定理一：

对于nim游戏的某个位置(x_1,x_2,x_3),当且仅当它各部分的nim-sum等于0时（即$x_1 \oplus x_2 \oplus x_3=0$），则当前位于必败点。

定理一也适用于更多堆的情况～

定理一的证明……

$$\begin{array}{rcl} 13 &=& 1101_2 \\ 12 &=& 1100_2 \\ 8 &=& 1000_2 \\ \hline \text{nim-sum} &=& 1001_2 = 9 \end{array}$$

思考（1）：

- 有了定理一，如果判断某个游戏的先手是输还是赢？

思考（2）：

- 对于必胜点，如何判断有几种可行的操作方案？

$$
\begin{array}{rcl}
13 & = & 1101_2 \\
12 & = & 1100_2 \\
8 & = & 1000_2 \\
\hline
\text{nim-sum} & = & 1001_2 = 9
\end{array}
$$

实例分析(HDOJ_1850)

- **Being a Good Boy in Spring Festival**

第三部分

Graph Games &
Sprague-Grundy Function

What is the graph game ?

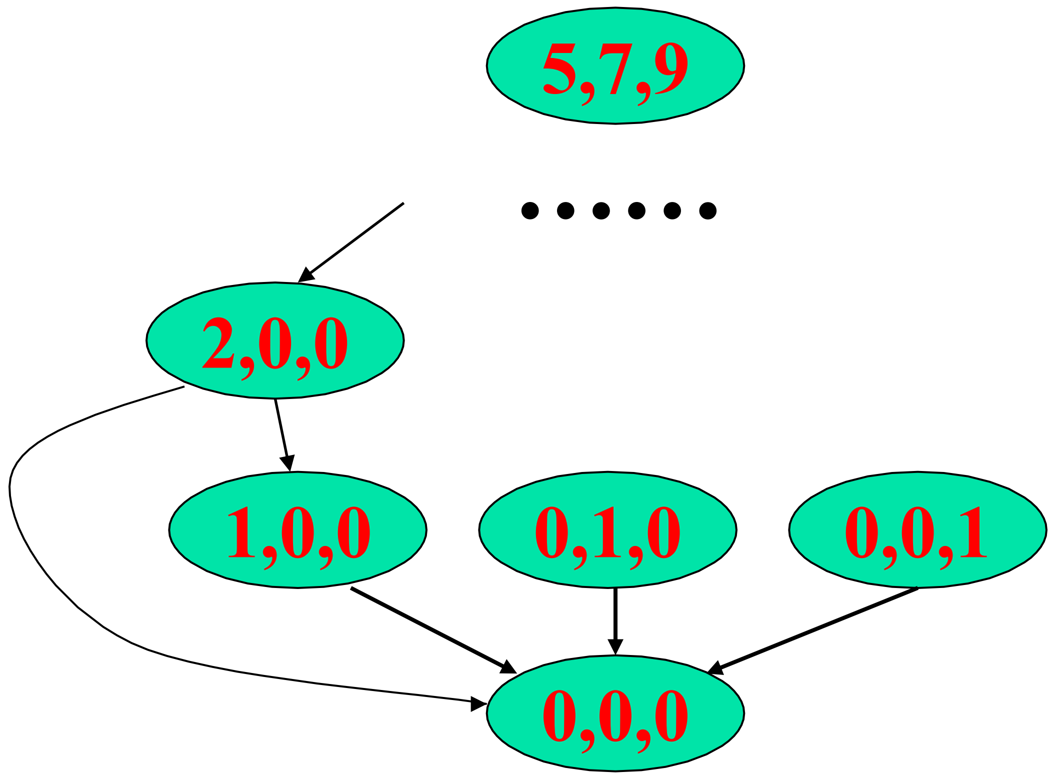
(1) Player I moves first, starting at x0.

(2) Players alternate moves.

(3) At position x, the player whose turn it is to move chooses a position $y \in F(x)$.

(4) The player who is confronted with a terminal position at his turn, and thus cannot move, loses.

Example about graph game:



The Sprague-Grundy Function.

Definition: *The Sprague-Grundy function of a graph, (X,F), is a function, g, defined on X and taking non-negative integer values, such that*

$$g(x) = \min\{n \geq 0 : n <> g(y) \text{ for } y \in F(x)\}. \quad (1)$$

In words, g(x) the smallest non-negative integer not found among the Sprague-Grundy values of the followers of x.

$$g(x) = \mathrm{mex}\{g(y) : y \in F(x)\}. \quad (2)$$

Use of the Sprague-Grundy Function:

P-positions: Positions x for which g(x) = 0

N-positions: Positions x for which g(x) > 0

Exercise:

- ***What is the SG-value of the subtraction game with subtraction set S = {1, 2, 3}?***

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	. . .
$g(x)$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	. . .

Question:

What can the S-G value describe?

Part 4:

Sums of Combinatorial Games

What ***is so-called ——***

"Sums of Combinatorial Games"?

Theorem 2.

If g_i is the Sprague-Grundy function of G_i
,
$i = 1, \ldots, n$, then
$G = G_1 + \cdots + G_n$ has Sprague-Grundy function
$g(x_1, \ldots, x_n) = g1(x_1) \oplus \cdots \oplus gn(x_n)$.

Applications:

Sums of three Subtraction Games.

In the first game:

$m = 3$ and the pile has 9 chips.

In the second:

$m = 5$ and the pile has 10 chips.

In the third:

$m = 7$ and the pile has 14c hips.

$g(9, 10, 14) =?$

附:参考源码(HDOJ-1536)

```
#include<stdio.h>
#include<string.h>
#include<algorithm>
using namespace std;
int k,a[100],f[10001];
int mex(int p)
{  int i,t;
    bool g[101]={0};
    for(i=0;i<k;i++)
    {
        t=p-a[i];
        if(t<0)
            break;
        if(f[t]==-1)
            f[t]=mex(t);
        g[f[t]]=1;
    }
    for(i=0;;i++)
    {
        if(!g[i])
            return i;
    }
}
```

```
int main()
{
    int n,i,m,t,s;
    while(scanf("%d",&k),k)
    {
        for(i=0;i<k;i++)
            scanf("%d",&a[i]);
        sort(a,a+k);
        memset(f,-1,sizeof(f));
        f[0]=0;
        scanf("%d",&n);
        while(n--)
        {
            scanf("%d",&m);
            s=0;
            while(m--)
            {
                scanf("%d",&t);
                if(f[t]==-1)
                    f[t]=mex(t);
                s=s^f[t];
            }
            if(s==0)
                printf("L");
            else
                printf("W");
        }
        printf("\n");
    }
}
```

课后练习

- 2008《ACM Programming》Exercise(12)_博弈入门
- 1517 A Multiplication Game
- 1079 Calendar Game
- 2147 kiki's game
- 1404 Digital Deletions
- 1536 S-Nim
- 1729 Stone Game
- 1730 Northcott Game
- 1760 A New Tetris Game
- 1809 A New Tetris Game(2)
- 1524 A Chess Game

记住:

- 学习是快乐的~

Welcome to HDOJ

Thank You ~