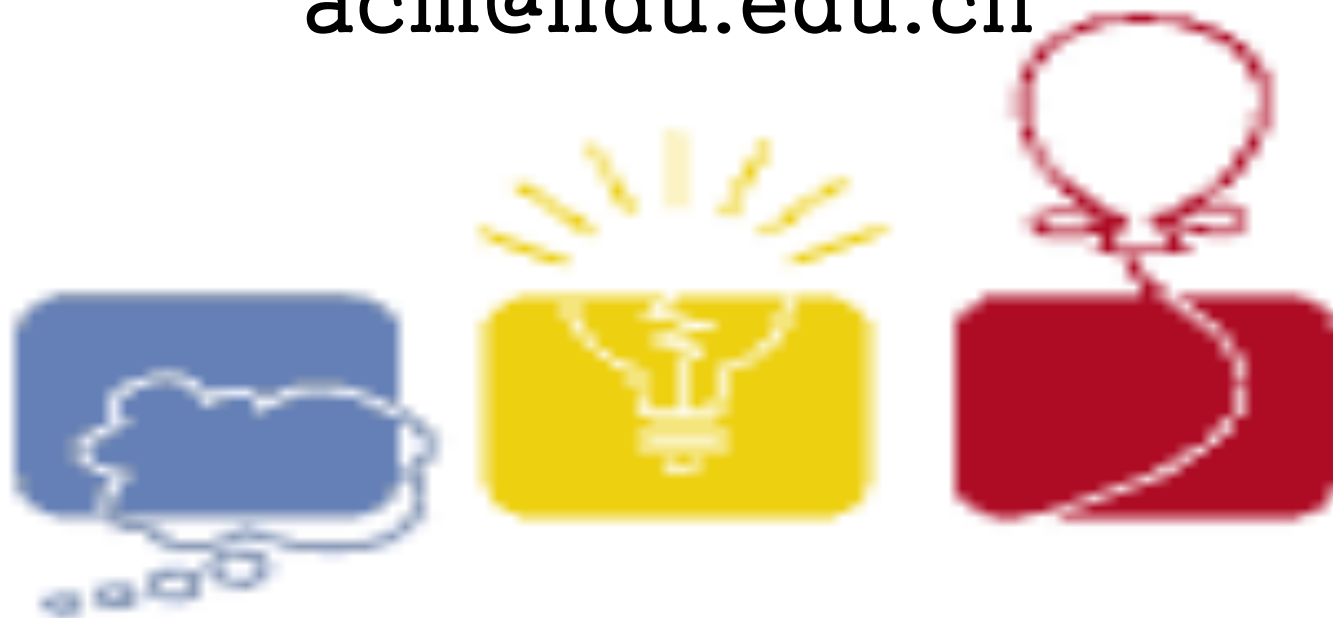
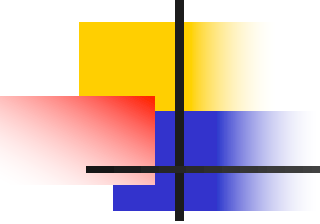


ACM程序设计

杭州电子科技大学 刘春英

acm@hdu.edu.cn





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(Previous 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:

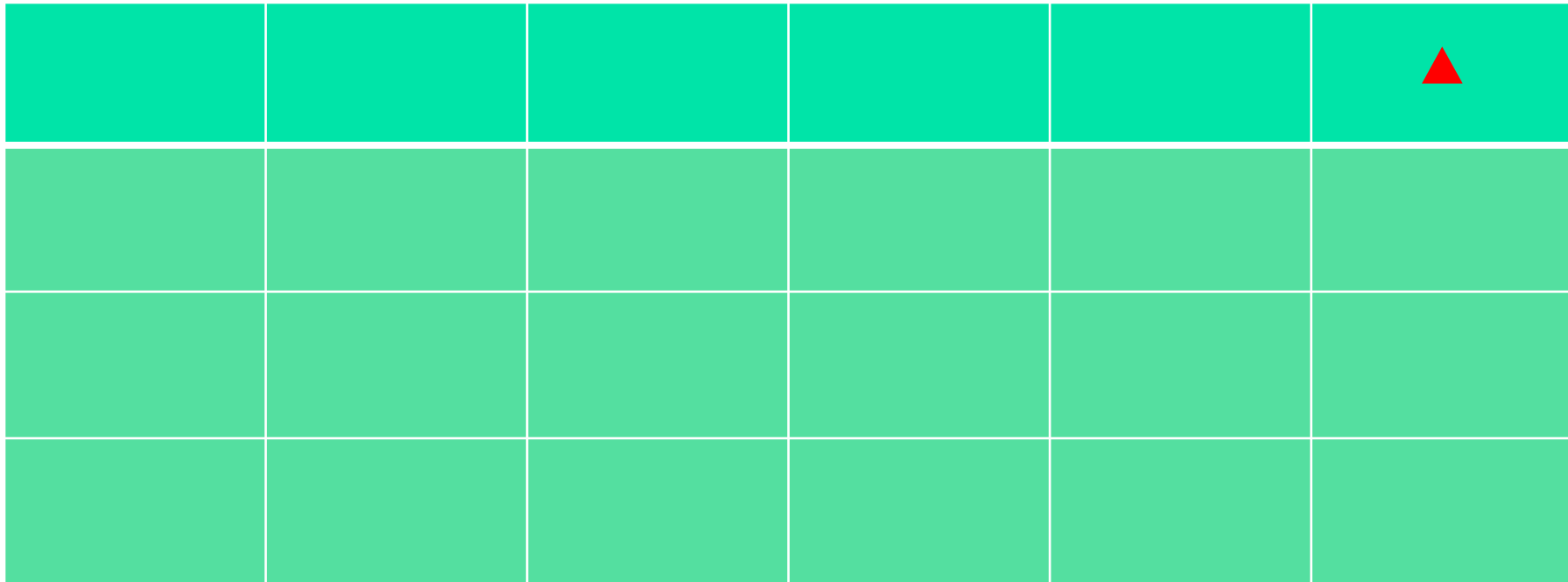
■ 取子游戏: $S = \{1, 3, 4\}$

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

■ 玩游戏的小男孩





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：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；





初步分析

- | | |
|------------------|--------------|
| ■ $(0, 0, 0)$ | ■ P-position |
| ■ $(0, 0, x)$ | ■ N-position |
| ■ $(0, 1, 1)$ | ■ P-position |
| ■ $(0, k, k)$ | ■ P-position |
| ■ $(14, 35, 46)$ | ■ ??? |



引入概念: Nim-Sum

- **定义:** 假设 $(x_m \cdots x_0)_2$ 和 $(y_m \cdots y_0)_2$ 的nim-sum 是 $(z_m \cdots z_0)_2$, 则表示成:

$$(x_m \cdots x_0)_2 \oplus (y_m \cdots y_0)_2 = (z_m \cdots z_0)_2,$$

- **其中**, $z_k = x_k + y_k \pmod{2} \quad (k=0 \cdots m)$.

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



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，如何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？

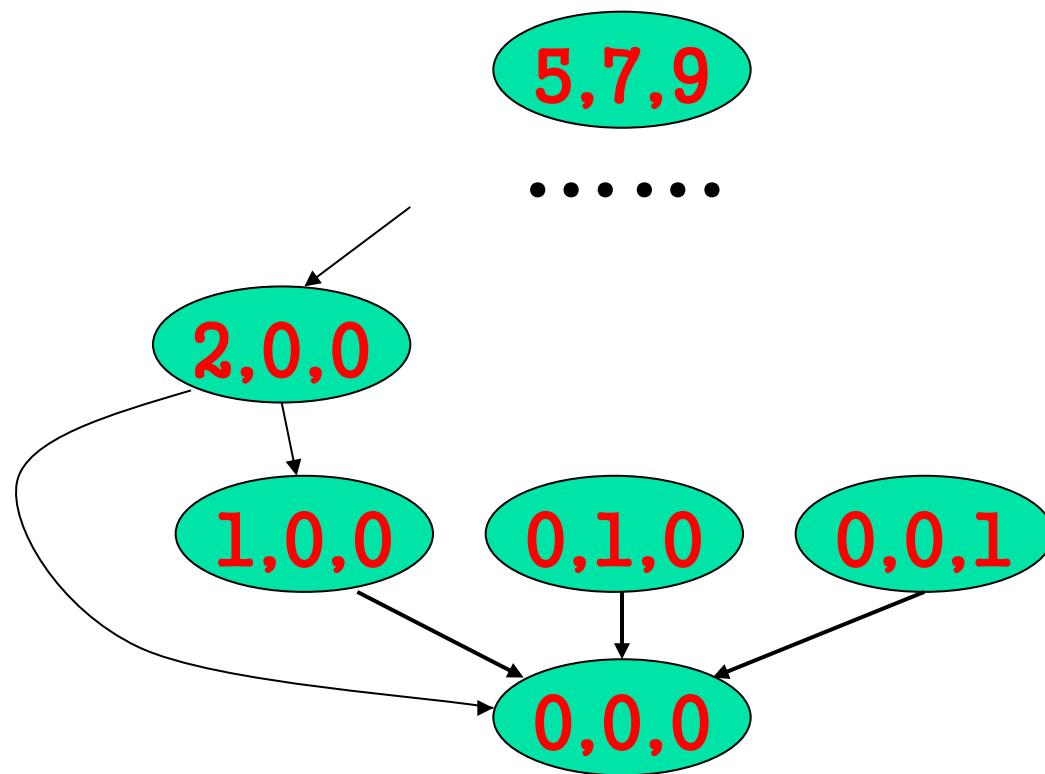
$$\begin{array}{rcl} 13 & = & 1101_2 \\ 12 & = & 1100_2 \\ 8 & = & 1000_2 \\ \hline \text{nim-sum} & = & 1001_2 = 9 \end{array}$$



第三部分

Graph Games & Sprague–Grundy Function

NIM游戏的状态转移图





SG函数

$$g(x) = \min\{n \geq 0 : n \neq g(y) \text{ for } y \in F(x)\}$$

也就是说，X节点的SG值是去除X的后继节点的SG值后的最小的非负整数。



SG函数的使用

必败点: 当节点 x 的 $sg(x) = 0$
必胜点: 当节点 x 的 $sg(x) > 0$



课内练习

- 若 $S=\{1,2,3\}$ ，请计算简单取子游戏的SG值。

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14...
<i>sg(x)</i>	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2...



疑问：

SG函数还能解决什么问题呢？



第四部分

组合游戏的并



例1

有一个2人小游戏定义如下：

有三堆扑克牌，分别为5,7,9张；

双方轮流取牌，每次可以选择任意一堆牌取走**1~3张**；

最后取牌的一方获胜。

请问，先手的人是输还是赢呢？

$g(5,7,9) = ?$



定理二

如果图游戏G由若干子图游戏 G_i 组成，即：

$G = G_1 + \dots + G_n$ ，假设 g_i 是 G_i ($i = 1, \dots, n$) 的SG函数值，那么，图游戏G的SG值计算如下：

$$g(x_1, \dots, x_n) = g_1(x_1) \oplus \dots \oplus g_n(x_n)$$

再看例1： $g(5, 7, 9) = ?$



例2

样例输入：

2 2 5

3

2 5 12

3 2 4 7

4 2 3 7 12

5 1 2 3 4 5

3

2 5 12

3 2 4 7

4 2 3 7 12

0

样例输出：

LWW

WWL

题目描述（略）

数据含义，请注意听讲

例2-参考代码

说明：

1、本参考代码的sg函数采用了普通DFS的实现模式；

2、可以在此基础上，改成**记忆化DFS**的实现方式；

3、也可以采用**非递归**的模式，从小到大计算；

4、不管是记忆化DFS，还是非递归模式，都需要一个额外的用来保存sg值的数组。

//普通DFS，效率低

```
#include<bits/stdc++.h>
using namespace std;
```

```
int k,a[100],f[10001];
```

```
int sg(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]=sg(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); //如果删除呢?
        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]=sg(t);
                s=s^f[t];
            }
            if(s==0)
                printf("L");
            else
                printf("W");
        }
        printf("\n");
    }
}
```



下周见

努力学习
实力报国

