Game theory

Impartial Games

Combinatorial games

- Two-person games with perfect information
- No chance moves
- A win-or-lose outcome
- **Impartial games**
  - Set of moves available from any given position is the same for both players
- **partizan games**
  - Each player has a different set of possible moves from a given position
  - Eg: Chess

Take away games – an example

- We start with ‘n’ coins, in each turn the player can take away 1, 3 or 4 coins
- Question:
  - What is the value of ‘n’ for which the first player can win the game if both the players play optimally ??

Solution for example

- L – Losing position
- W – Winning position

L	W	L	W	W	W	W	L	W	L	W	W	W	W	L	W
0	1	2	3	4	5	6	7	8	9	10	1	12	13	14	15

- 0,2,7,9,14,16,… are losing positions
- 7N,7N+2 are losing positions for N>=0

Properties of positions

- All terminal positions are losing.
- If a player is able to move to a losing position then it is a winning position.
- If a player is able to move only to the winning positions then it is a losing position.

WL – Algorithm

```
boolean isWinning(position pos) {
    moves[] = possible positions to which we
              can move from the position pos;
    for (all x in moves)
        if (!isWinning(x))
            /* can move to Losing pos */
            return true;
    return false;
}
```

Try this problem

- http://www.spoj.pl/problems/NGM/
- http://www.spoj.pl/problems/MCOINS/

Game of Nim

- There are ‘n’ piles of coins.
- In each turn a player chooses one pile and takes at least one coin from it. If a player is unable to move he loses.

Solution of Nim Game

- Let n1, n2, … nk, be the sizes of the piles. It is a losing position for the player whose turn it is if and only if $\mathbf{n_1}$ **xor** $\mathbf{n_2}$ **xor .. xor** $\mathbf{n_k} = \mathbf{0}$.

- **Why does it work ??**

Why does it work??

- From the losing positions we can move only to the winning ones:
  - if xor of the sizes of the piles is 0 then it will be changed after our move
- From the winning positions it is possible to move to at least one losing:
  - if xor of the sizes of the piles is not 0 we can change it to 0 by finding the left most column where the number of 1s is odd, changing one of them to 0 and then by changing 0s or 1s on the right side of it to gain even number of 1s in every column.

Composite Games – Grundy numbers

```
int grundyNumber(position pos) {
    moves[] = possible positions to which I can
                move from pos set s;

    for (all x in moves)
        insert into s grundyNumber(x);
    //return the smallest non-negative integer
    //not in the set s;

    int ret=0;
    while (s.contains(ret)) ret++; return ret;

}
```

Practice problems

- http://www.spoj.pl/problems/QCJ3/
- http://www.spoj.pl/problems/RESN04/
- http://www.spoj.pl/problems/MMMGAME
- http://pclub.in/index.php/wpc-archives/16-kodefest-solutions/87-problem-e
- http://www.spoj.pl/problems/PEBBMOV/
- http://www.codechef.com/problems/CHEFBRO
- http://www.spoj.pl/problems/HUBULLU/
- SRM 330 DIV I Hard

- http://www.codechef.com/problems/BIGPIZA
- http://projecteuler.net/problem=301
- http://www.codeforces.com/contest/87/problem/C

- EASY DP: http://www.spoj.pl/problems/CRSCNTRY

References

- http://www.math.ucla.edu/~tom/Game_Theory/comb.pdf
- http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=algorithmGames