II Combinatorial Game theory Introduction

Introduction

CGT

① No move by change (No dice throw / coin throw)

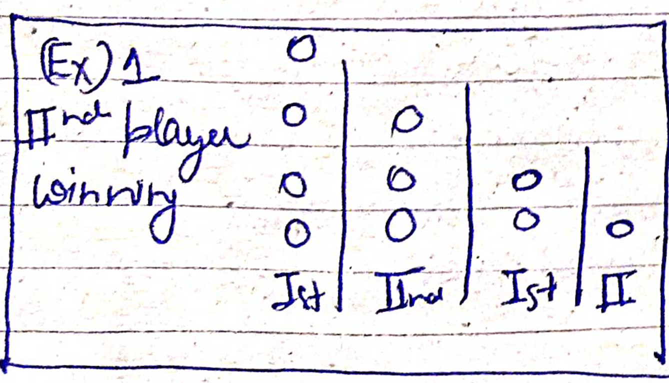
② Typically Two players taking turn one by one
(Sudoku exception)

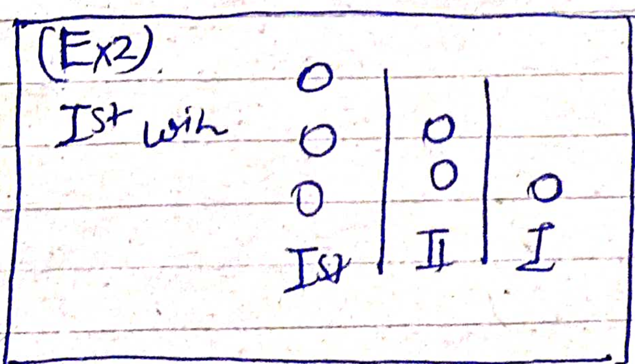
③ Perfect Information

④ Two types: (I) Impartial Game:
Both players have same available moves.
Ex: Nim.
(II) Both players have different available moves. Ex. Chess & Tic-Tac-Toe.
(Partisan Games)

Impartial Game Example

(1) Given a pile of (n) coins, each of the two players can pick a coin on its turn. The player that ~~match~~ makes the last move is the winner. Find the winner for given (n).

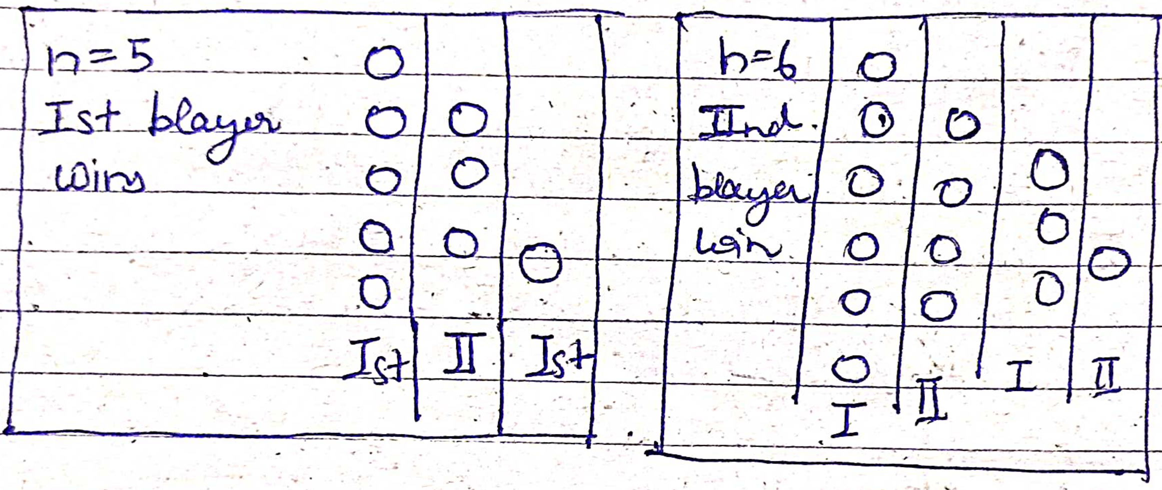




In this case if n → even
II wins

n → odd
Ist wins

(2) Each Player can either pick 1 or 2 and both player play optimally
(try to win)




```
Void winner (int n)
{ if (n%3 ==0)
        print ("II");
  else
        print ("I");
}
```

n	winner
1	I] 1st pick all
2	I
3	II] Ist can take only 1 & 2
4	I] Ist count 3
5	I
6	II] Ist end 4 & 5
7	I] Ist count 6
8	I
9	II] Ist end 8, 7

③ Each Player can pick 1 to K coins.

I/p : n=6   K=2
O/P : II

I/p : n=7 , K=1
O/P : I

I/P. n=7 K=3
O/P: I

```
void findwinner(int n, int k)
{
if( n% (K+1) ==0)
    print("I");
else print("II");
}
```

① If n = 1, 2, ... K winner is Ist by picking all coins.

② If n = K+1 whatever Ist player pick, the game is going to reach in case ①.

③ If n = K+2, K+3 ... K+1+K the Ist player can make the game reach in ② case.

④ If n = 2(K+1) whatever Ist player pick, game to reach in case 3.

Grundy No. (Nimbers)
↳ When so set pattern to pick coins

I/P: $n = 15$
picks = {1, 2, 3}
O/P: Ist

I/P: $n = 7$
picks [] = {5, 1, 4}
Op: Ist

I/P: $n = 8$
picks = {5, 1, 4}
O/P: IInd.

Mex and Grundy Numbers

① Mex({Set of Non-Negative Integers})
= Smallest Non-negative Int missing from the set

mex ({0, 2, 3}) = 1
mex ({1, 4, 5}) = 0
mex ({0, 1, 4, 2}) = 3
mex ({3, 2, 1, 0}) = 4

Logic of working Grundy:
If a player cannot reach state with 0 coins from current state, the player is going to loose the game.

② Grundy Number (Nimber)

⇒ Grundy No. tell us the winner from current state of game.

⇒ If the value is greater than 0, the current player wins. Else current player looses.

⇒ Grundy (n) = Mex({Grundy(n-pick[0]), Grundy (n-pick[1]) , ...})

Grundy(0) = 0

Grundy(1) = Mex({Grundy(0)}) = Mex({0}) = 1

Grundy (2) = Mex(Grundy (1), Grundy (0))
= Mex (0,1) = 2

(Ex) n=5
pick = {1,2,3}

Grundy (3) = Mex (Grundy (2), Grundy (1), Grund(0))
= Mex (2, 0, 1) = 3

Grund(4) = Mex (G(3), G(2), G(1))
= Mex (3, 2, 1) = 0

Grundy(5) = Mex (G(4) G(3) G(2))
= Mex (0, 3, 2) = 1

0
0
0
0
0

Grundy Number implementation

```
int calculateMex (set<int>& s){
    int Mex=0;
  while (s.contain(Mex)){
        Mex++;}
  return Mex;
}

int calculateGrundy (int N, int pile[], int size){
        if ( n==0) return 0;
        set <int> s ;

       for (int i=0; i<size; i++)
         { if (n - pick[i] >=0)
             { S.insert (calgrundl(n-pick[i], [illegible], pile, size));
             }
       }

    return calculateMex (S);
}
```

TC → O(size^N)
SC → O(N)

Apply DP → TC → O(N²)
SC → O → O(N²)

Composite Games

- There are two piles
- A player on its turn can pick 1, 2 or 3 coins from any of the two piles.
- Find the winner.

I/p: $n_1 = 4$ $n_2 = 3$
O/p = 1st

I/p: $n_1 = 2$ $n_2 = 2$
O/p: 2nd

I/p: $n_1 = 1$ $n_2 = 5$
O/p: 2nd

I/p: $n_1 = 2$ $n_2 = 3$
O/p: 1st

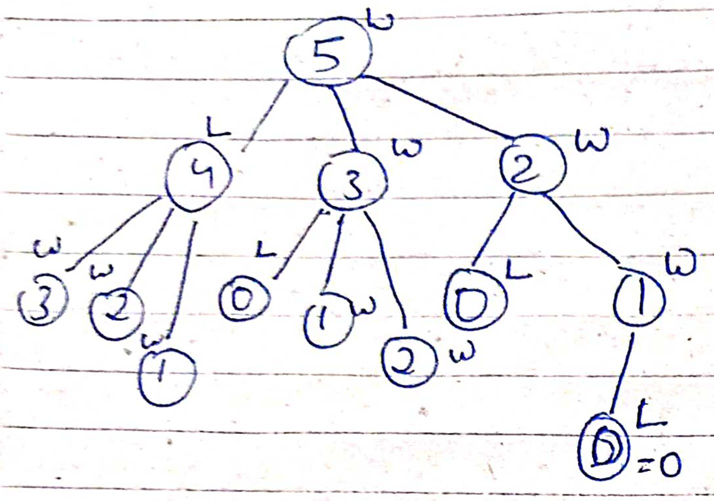
```
string findWinner (int n1, int n2)
{ if (grundy (n1) == grundy (n2))
      return "2nd";
  else return "1st";
}
```

① $n_1 = 1$ $n_2 = 5$
grundy(1) = 1
grundy(5) = 1
2nd win.

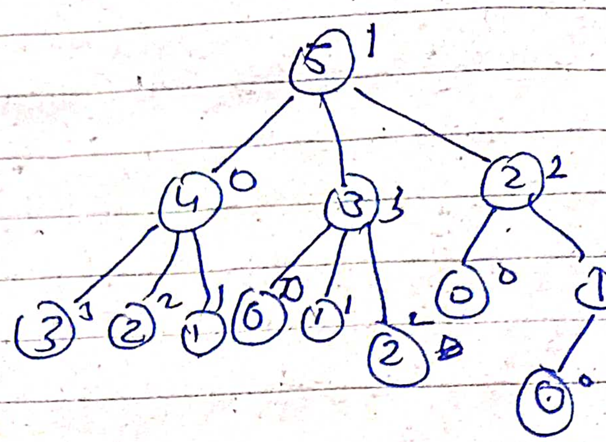
derived from sprague grundy theorem.
if grund(n_1) == grund(n_2) = K

$K \oplus K = 0 \rightarrow$ 2nd Win

Win/Lose state



Grundy



Sprague Grundy Theorem

* Three Piles
* Each Player on its turn can pick 1, 2, or 3 coins from a pile.

I/P: $n_1 = 1$, $n_2 = 1$, $n_3 = 1$
O/P: 1st

I/P: $n_1 = 1$, $n_2 = 4$, $n_3 = 1$
O/P: 2nd

I/P: $n_1 = 1$, $n_2 = 2$, $n_3 = 3$
O/P: 2nd

I/P ⇒ $n_1 = 1$, $n_2 = 4$, $n_3 = 5$
O/P = 2nd.

I/P: $n_1 = 1$, $n_2 = 3$, $n_3 = 5$
O/P: 1st

Sprauge Grundy theorem

Find Grundy Number (or Nimber) of any composite game which is a combination of multiple simple combinatorial impartial games.

Grundy Number of Composite Game =
$Grundy(g_1) \wedge Grundy(g_2) \ldots \wedge Grundy(g_n)$

$\wedge$:- Bitwise XOR
$g_1, g_2 \ldots g_n$:- n games.

① $n_1 = 1$ $n_2 = 3$ $n_3 = 5$

grundy = grund (1) ^ grundy (3) ^ grundy (5)
= 1 ^ 3 ^ 1
= 3 → Ist

② $n_1 = 1$, $n_2 = 2$, $n_3 = 3$
grundy = G(1) ^ G(2) ^ G(3)
= 1 ^ 2 ^ 3
= 0 → IInd

The Game of Nim

- n Piles with possibly diff no. of coins.
- A player on its turn can take any no. of coins from a single pile
- The player who picks the last coins(s) is winner.

I/p: nim[] = {1, 2, 3}
O/p: 2nd

I/p: nim[] = {2, 3, 4}
O/p: 1st

$1 \wedge 2 \wedge 3 = 0 \rightarrow \text{II}$

$2 \wedge 3 \wedge 4 > 0 \rightarrow \text{I}$

nim-sum = nim[0] ^ nim[1] ^ ... nim[n-1]

It is grundy number only for game of nim.

SGT → but for nim a game XOR all element.