0, 1, 2, 3, 1, 4, 3, 2, 1, 4, 2, 6, 4, 1, 2, 7, 1, 4, 3, 2, 1, 4, 6, 7, 4, 1, 2, 8, 5, 4, 7, 2, 1, 8, 6, 7

$\sqcup = \{\ \} \qquad *0$

$x = \{\sqcup\} \approx \{*0\} \qquad *1$

$x \ \ x = \{x\} \qquad *0$
$\phantom{x \ \ x = \{}*1$

$xx = \{x, \sqcup\} \qquad *2$
$\phantom{xx = \{}*1, *0$

$xxx = \{xx, x \ \ x, x\} \qquad *3$
$\phantom{xxx = \{}*2 \quad *0 \quad *1$

$xxxx = \{xxx, x \ xx, xx, x \ \ x\} \qquad *1$
$*3 \quad *1 \oplus *2 = *3 \quad *2 \quad *1 \oplus *1 = *0$

$xxxxx = \{xxxx, xxx \ \ x, xx \ \ xx, xxx, x \ \ xx\} \qquad *4$
$*1 \quad *3 \oplus *1 = *2 \quad *0 \quad *3 \quad *1 \oplus *2 = *3$

01
10
$\overline{11}$

$xxxxxx = \{xxxxx, xxxx, x \ \ xxxx, x \ \ xxx, xx \ \ xxx, xx \ \ xx\} \ *3$
$*4 \quad *1 \quad *1 \oplus *1 = *0 \quad *1 \oplus *3 = *2 \quad *2 \oplus *3 = *1 \quad *2 \oplus *2 = *0$

$\text{mex}(\{4, 1, 0, 2, 1, 0\}) = 3$

$\underset{*4 \to *0}{\overset{9}{xxxx \, x \, xxxx}} + \underset{*3}{\overset{6}{xxxxxx}} + \underset{*1}{\overset{1}{x}} + \underset{*2}{\overset{7}{xxxxxxx}}$

$*1 + *1 + *3 + *1 + *2$

$\approx *(1 \oplus 1 \oplus 3 \oplus 1 \oplus 2)$

$= *0$

0 0 0	
1 0 0	4
0 1 1	3
0 0 1	1
0 1 0	2
1 0 0	

$\ \ 100$
$\oplus\ 100$
$\overline{\ \ 000}$

$*2$
$*1 \oplus *3$

$xx\cancel{x}xxx \qquad xxxxxxxxxxxx \qquad xxxx \ xxxxx$
$*3 \qquad *6 \qquad *4$

010			
011	3 → 2		
110	6	→ 7	111
⊕ 100	4	→ 5	101
001			

0 5 10 15

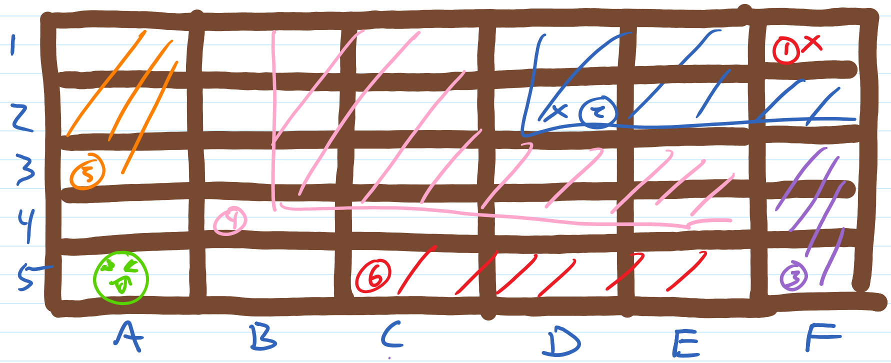
0, 1, 2, 3, 1, 4, 3, 2, 1, 4, 2, 6, 4, 1, 2, 7, 1, 4, 3, 2, 1, 4, 6, 7, 4, 1, 2, 8, 5, 4, 7, 2, 1, 8, 6, 7

Chomp

Play on m×n grid. Take turns selecting remaing cell, remove all above and to right

Last move loses.

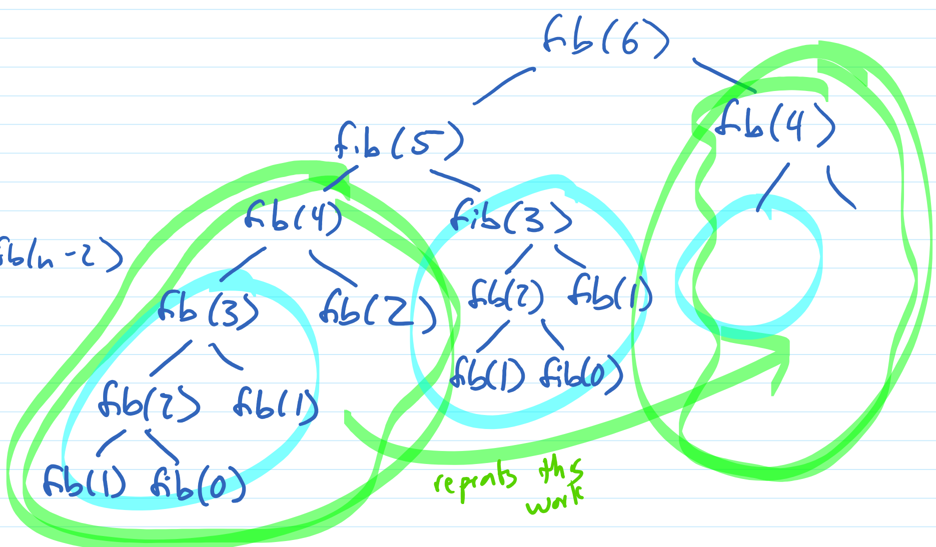
misère



outcome_class(p)

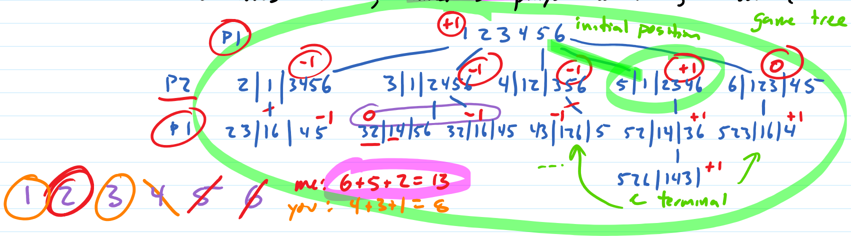
```
if p is end of game                          terminal
    return value according to rules  — normal  P (lose)
                                     — misère  N (win)
else
    S ← positions reachable in 1 move from p
    if S contains a P position
        return N
    else
        return P
```

```
def fib(n):
    if n<2:
        return n
    else:
        return fib(n-1) + fib(n-2)
```



000
100
110
200
111
210
211
220
221
222

Divisors : Start with 1...n, players take turns taking a number with remaining divisors; opponent gets all the remaining divisors. Game is over when no moves remain; winner is player with higher sum (draw if =)



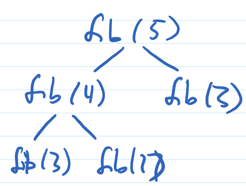
+1 P1 wins
0 draw
-1 P2 wins

Minimax(p)

if p is end of game (terminal)
  return value according to rules
else
  S ← positions reachable in 1 move from p (children)
  if P1 moves at p then return $\max_{s \in S}$ Minimax(s)
  else return $\min_{s \in S}$ Minimax(s)

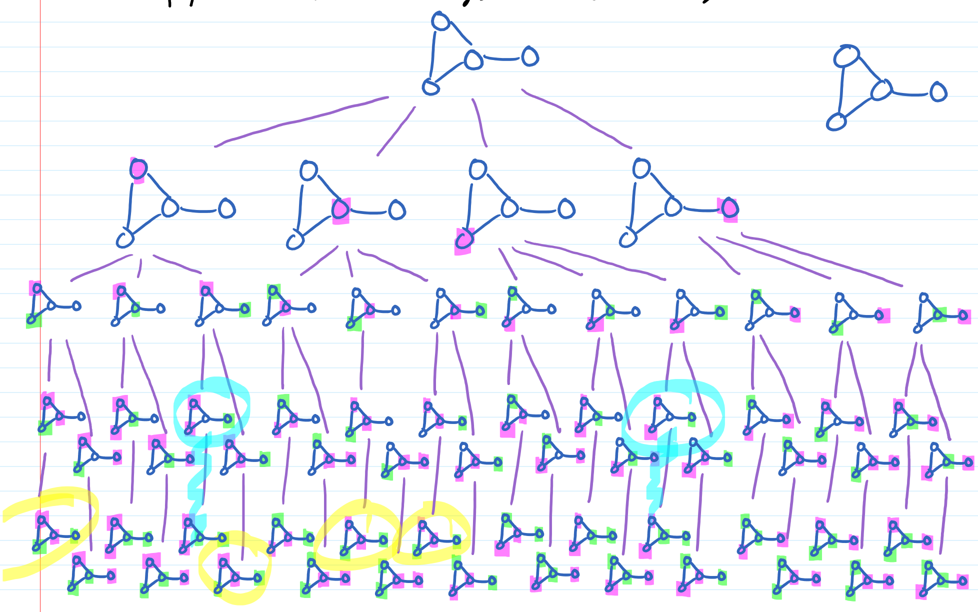
Minimax(p)
  if p is end of game
    return value according to rules
  else
    let S = positions reachable in 1 move from p
    if p is P1's turn
      return $\max_{p' \in S}$ Minimax(p')
    else
      return $\min_{p' \in S}$ Minimax(p')

def fib(n):
  if n<2: add (n,n) to memo
    return n
  else:
    return fib(n-1) + fib(n-2)   check if n-1, x is in memo; if so, use value in memo; check if (n-2, y) in memo



Graph Game

Graph: take turns coloring a vertex in a graph with your color
player who covers the most edges wins (draw if =)



Dynamic Programming

Order positions by maximum distance to end.

Determine winner of distance 0 positions (end) by

Use recursive formula to determine value of other positions in order of