# HKOI Training 2017
# Game Theory

Alex Tung

11 February 2017

# ‘Game’

- Not just any game!

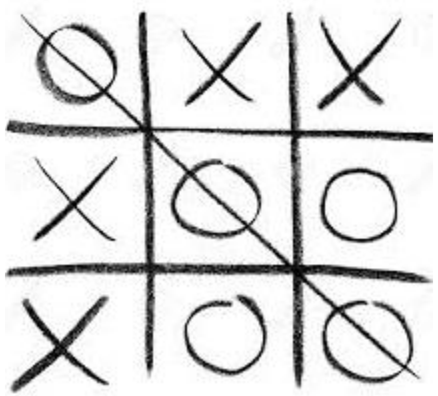
# So... what sort of games?

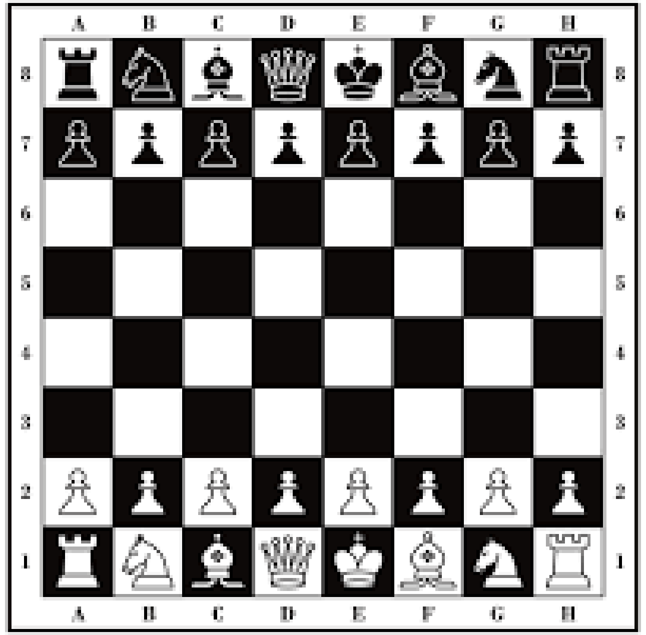
- Combinatorial games

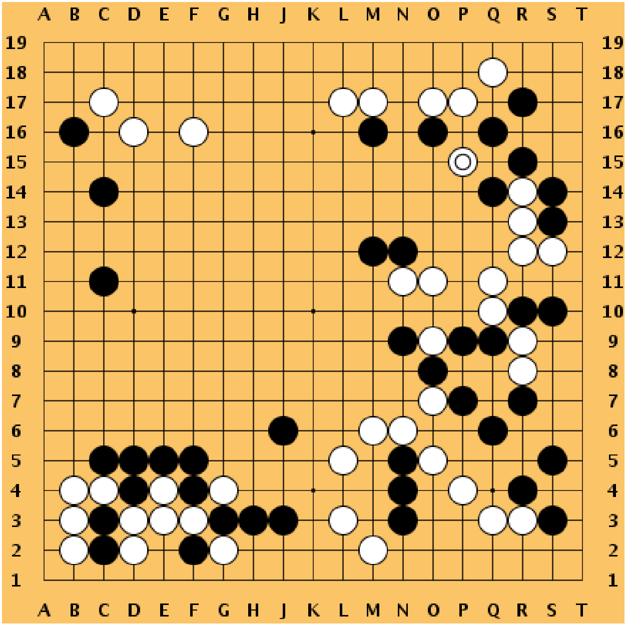
- Two players
- They **take turn** to change the **game state** with a **move**
- **Perfect information**; no chance/guesswork involved
- **Progressively finite**: the game must end after a finite sequence of moves

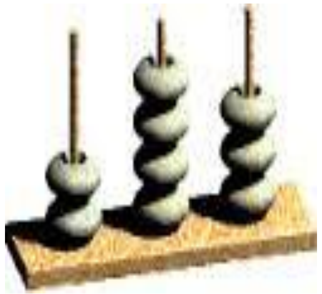
# Examples

- Tic-tac-toe, Chess, Go, Mancala, Nim





# Today's focus

- Impartial combinatorial game

- *Impartial*: at a particular position, both players have the same set of available moves
  - Nim is impartial
  - Mancala, Chess, Go, and Tic-tac-toe are not

# Normal and Misère game play

- There are (one or more) ending positions in the games we consider

- Normal: first to reach an ending position wins
- Misère: first to reach an ending position loses

- Unless otherwise specified, we assume the normal game play

# What it means to solve a game

- Find the outcome (win/draw/lose) **under perfect play**
- (Optional) Find the best possible moves at each position

# Game 1: Take-away game

- Game parameters: N, K

- We start with N stones
- In each move, a player can remove any number of stones from 1 to K (1, 2, ..., K)

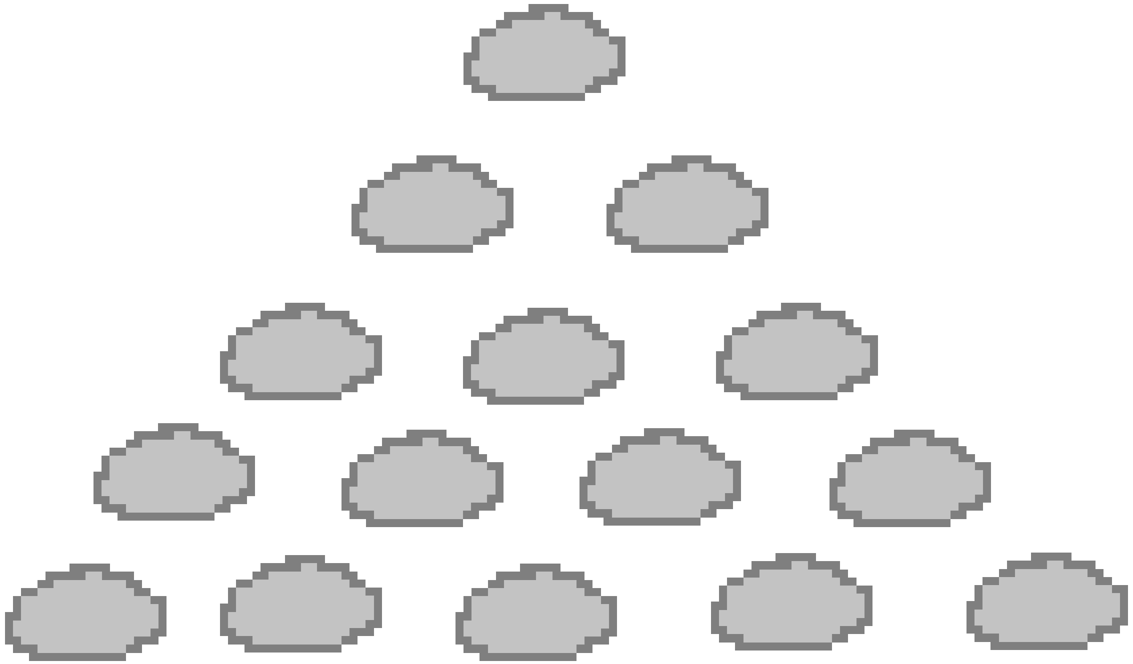
- Ending position: no stones left
- Who will win?

# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3

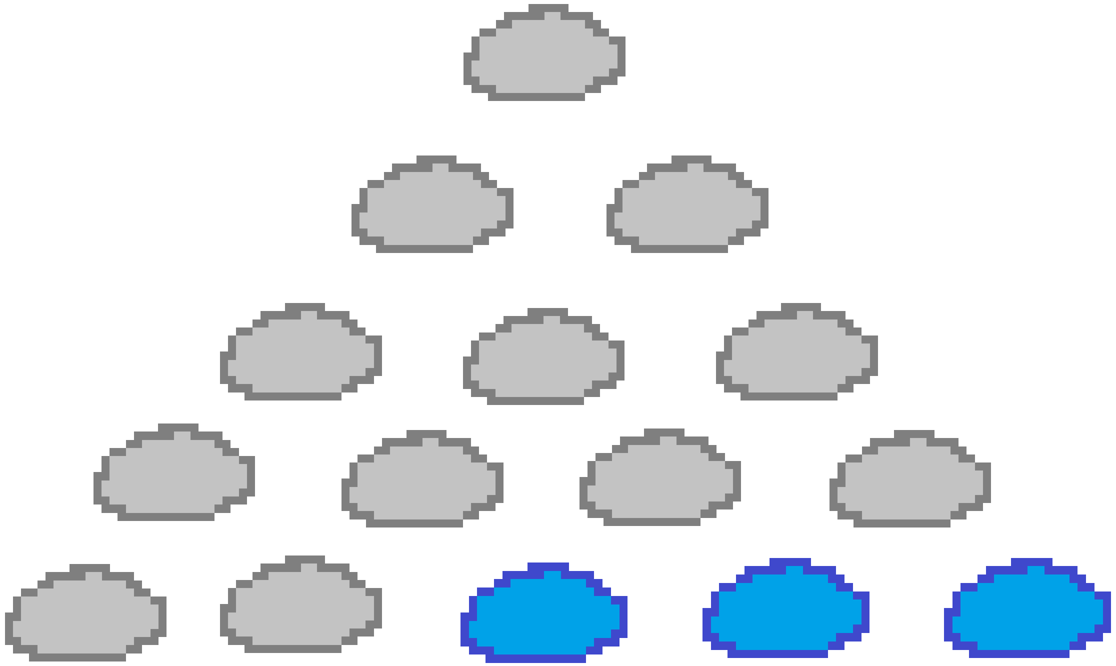


# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3

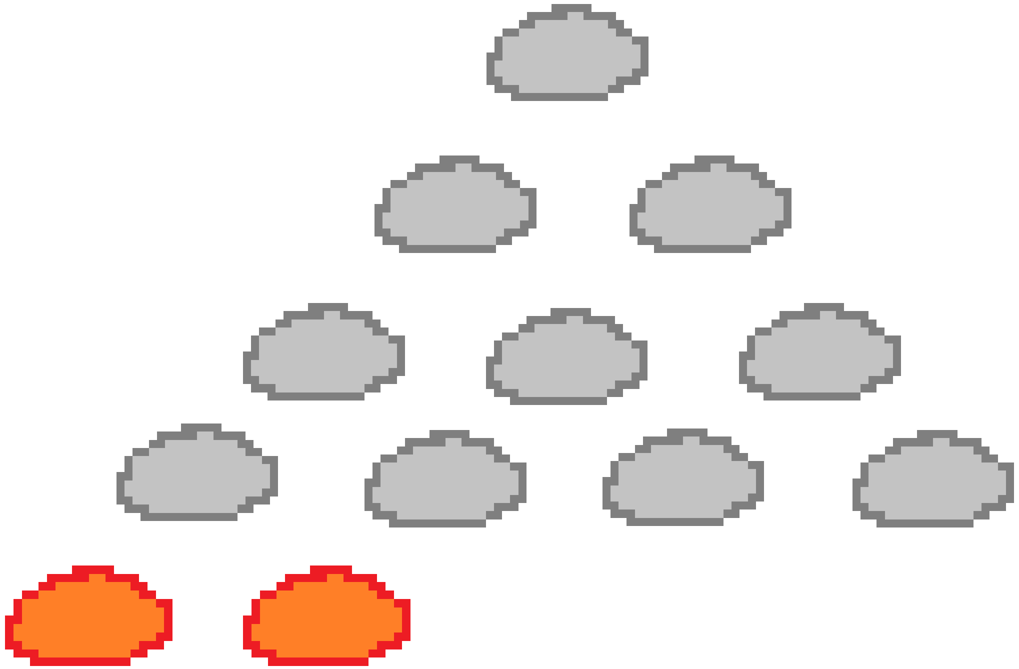


# Sample play

Player 1:

Player 2:

- N = 15, K = 3

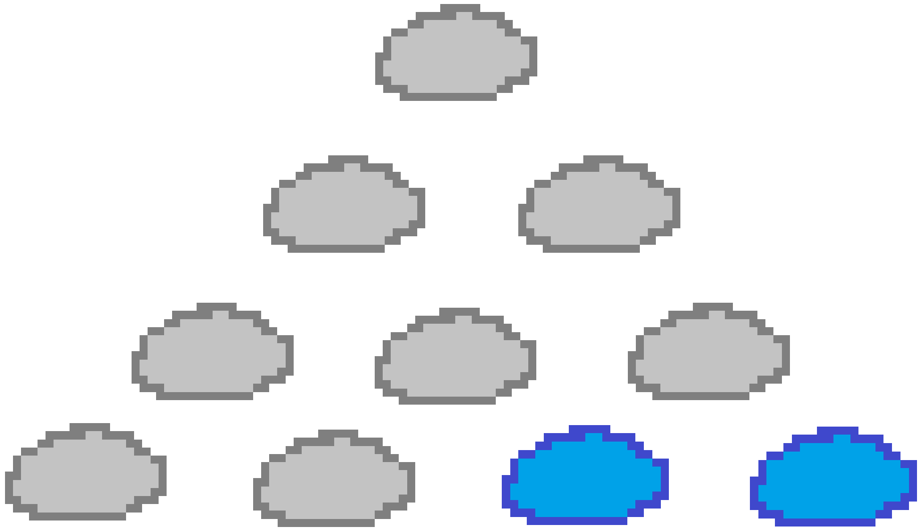


# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3

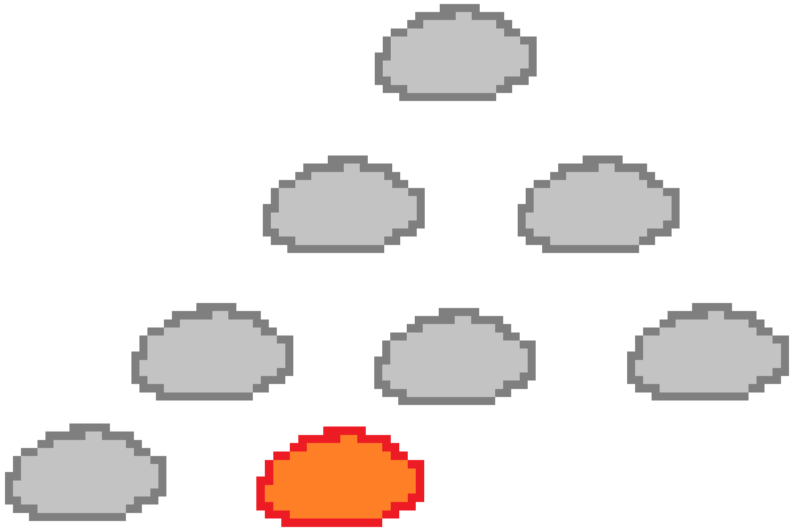


# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3

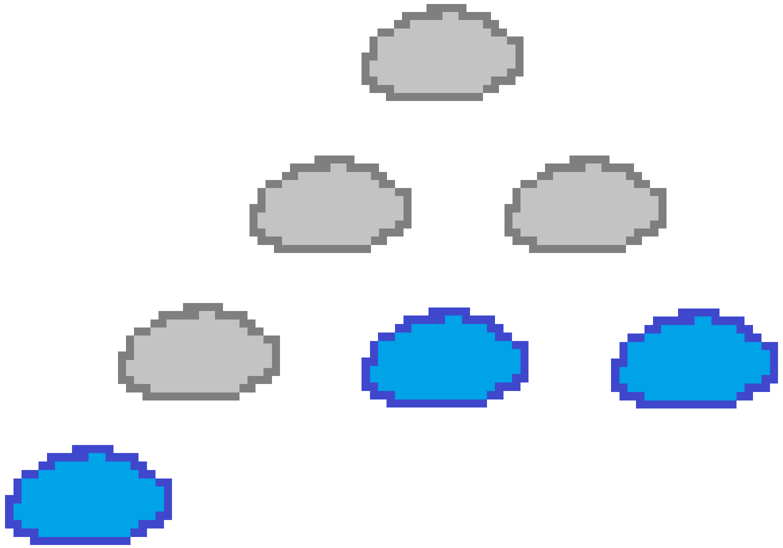


# Sample play

- N = 15, K = 3

Player 1: 

Player 2: 

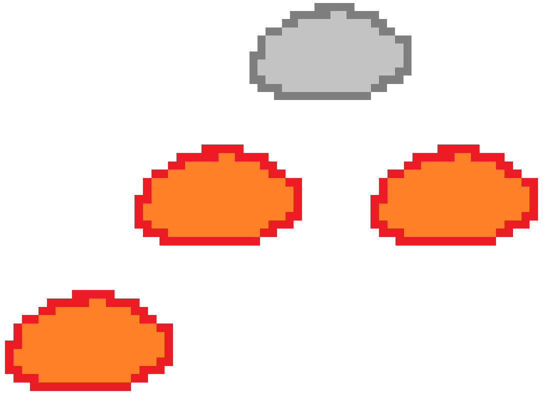


# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3



# Sample play

- N = 15, K = 3

Player 1: 

Player 2: 

# Sample play

Player 1: 

Player 2: 

- N = 15, K = 3

(To slide 22)

# Sad(?) fact

- Same as many (simple) combinatorial games, the result has been determined from the very beginning
- In this take-away game with (N, K) = (15, 3), player 1 wins (under perfect play)

# Solving for general (N, K)

- Fix K, try small N

- e.g. K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1/P2? | **P2** | P1 | P1 | P1 | P2 | P1 | P1 | P1 | P2 | P1 | P1 | P1 |

# Solution

- K = 3:

```
If N is a multiple of 4, player 2 wins.
Otherwise, player 1 wins.
```

- Much more general statement:

```
If N is a multiple of (K + 1), player 2 wins.
Otherwise, player 1 wins.
```

# Winning Strategy

- If N is a multiple of (K + 1)
  - Suppose player 1 takes X stones
  - Player 2 should take (K + 1 – X) stones next

- If N is not a multiple of (K + 1)
  - Player 1 should take N % (K + 1) stones
  - Then, use the strategy above

- Let’s watch sample play again :)

# P-position, N-position

- P-position: **P**revious player wins
- N-position: **N**ext player wins

- An ending position is a P-position

- If initial position is an N-position, player 1 wins
- If initial position is a P-position, player 2 wins

# Example

- Take-away game with K = 3
- Previously we use:

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1/P2? | P2 | P1 | P1 | P1 | P2 | P1 | P1 | P1 | P2 | P1 | P1 | P1 |

- Now we can use:

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | N | N |

# Is a given position P or N?

- Suppose we can reach positions $x_1$, $x_2$, $x_3$, ..., $x_k$ from the current position $X_{cur}$

If all $x_i$ are N-positions, $X_{cur}$ is a P-position.
Otherwise, $X_{cur}$ is an N-position.

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | | | | | | | | | | | |

0 is an ending position

>> P-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | | | | | | | | | | |

1 can reach 0 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | | | | | | | | | |

2 can reach 0 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | | | | | | | | |

3 can reach 0 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | | | | | | | |

4 cannot reach any P

>> P-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | | | | | | |

5 can reach 4 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | | | | | |

6 can reach 4 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | | | | |

7 can reach 4 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | | | |

8 cannot reach any P

>> P-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | | |

9 can reach 8 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | N | |

10 can reach 8 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | N | N |

11 can reach 8 (P)

>> N-position

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | N | N |

# Blank slide

- Rest / Q&A

# Game 2: Nim

- Game parameters: N, $a_1$, $a_2$, ..., $a_N$

- We start with N stone piles
- The i-th pile contains $a_i$ stones
- In each move, a player can choose any *non-empty* stone pile and remove some (> 0) number of stones from the pile

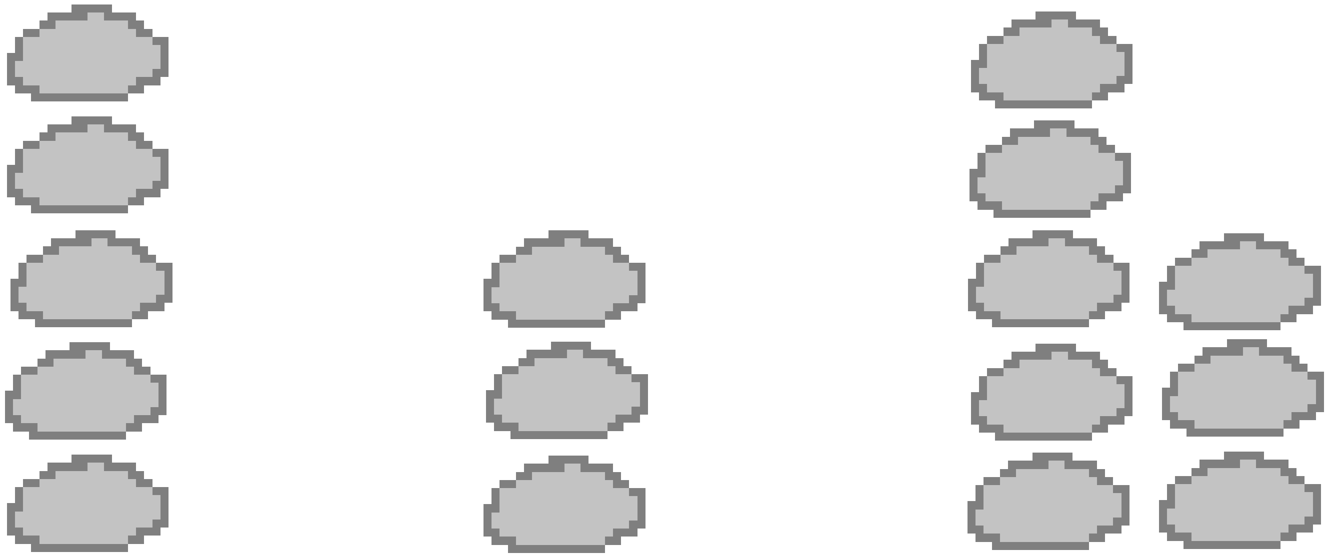
- Ending position: no stones left
- Who will win?

# Sample play

- $N = 3, a_1 = 5, a_2 = 3, a_3 = 8$

Player 1: 

Player 2: 

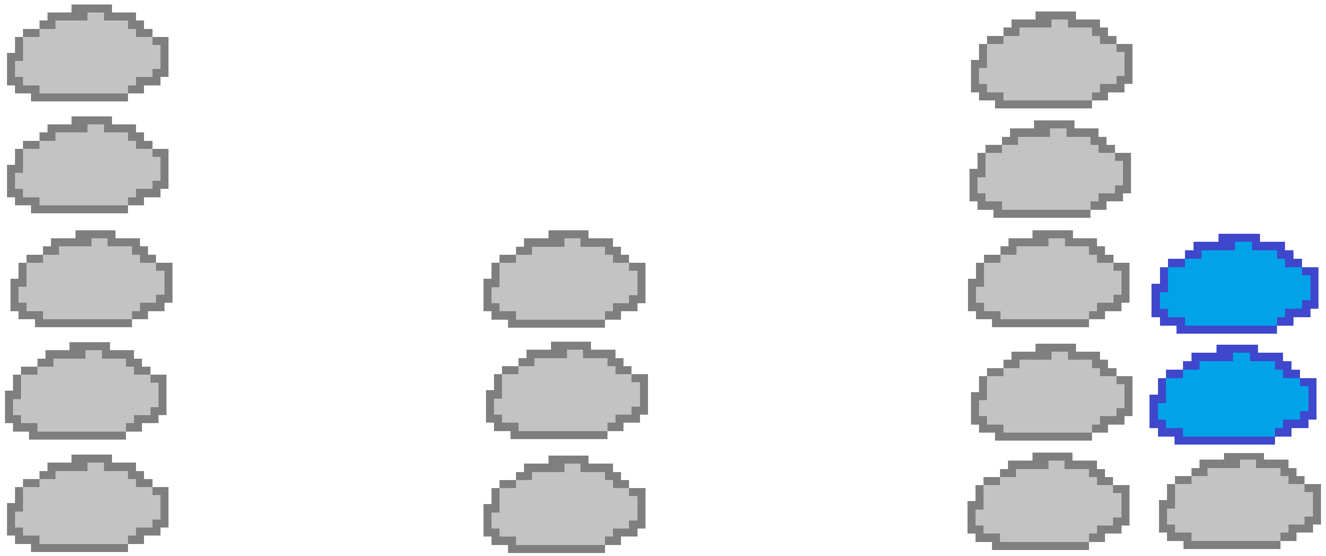


# Sample play

Player 1: 

Player 2: 

- $N = 3,\ a_1 = 5,\ a_2 = 3,\ a_3 = 8$

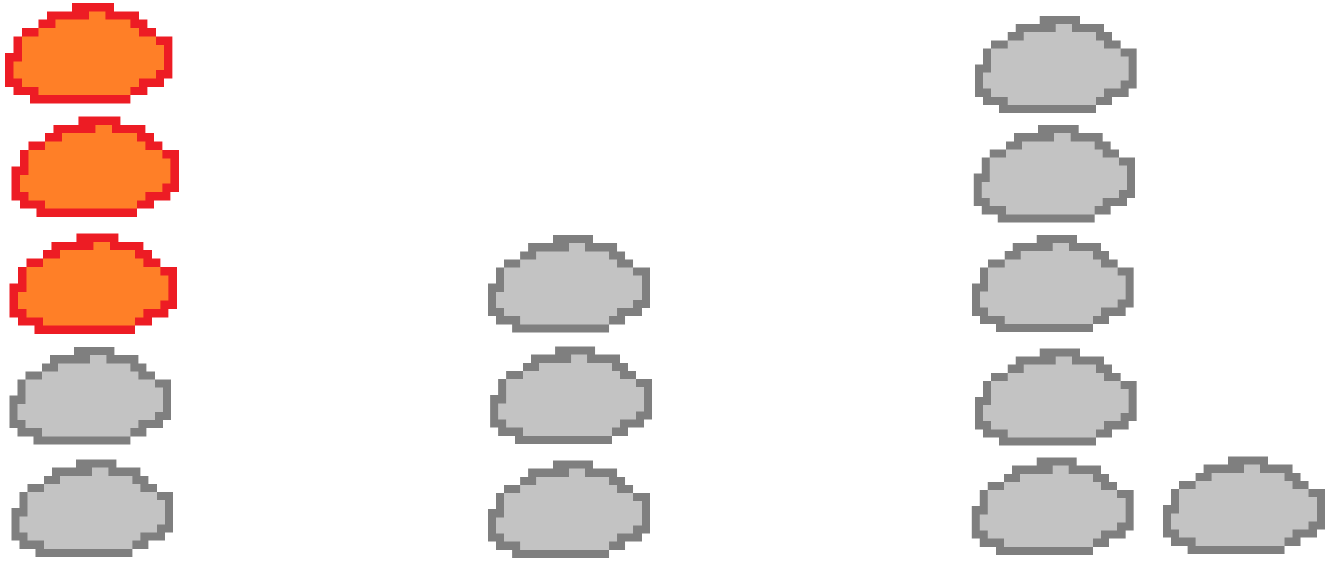


# Sample play

Player 1: 

Player 2: 

- $N = 3,\ a_1 = 5,\ a_2 = 3,\ a_3 = 8$

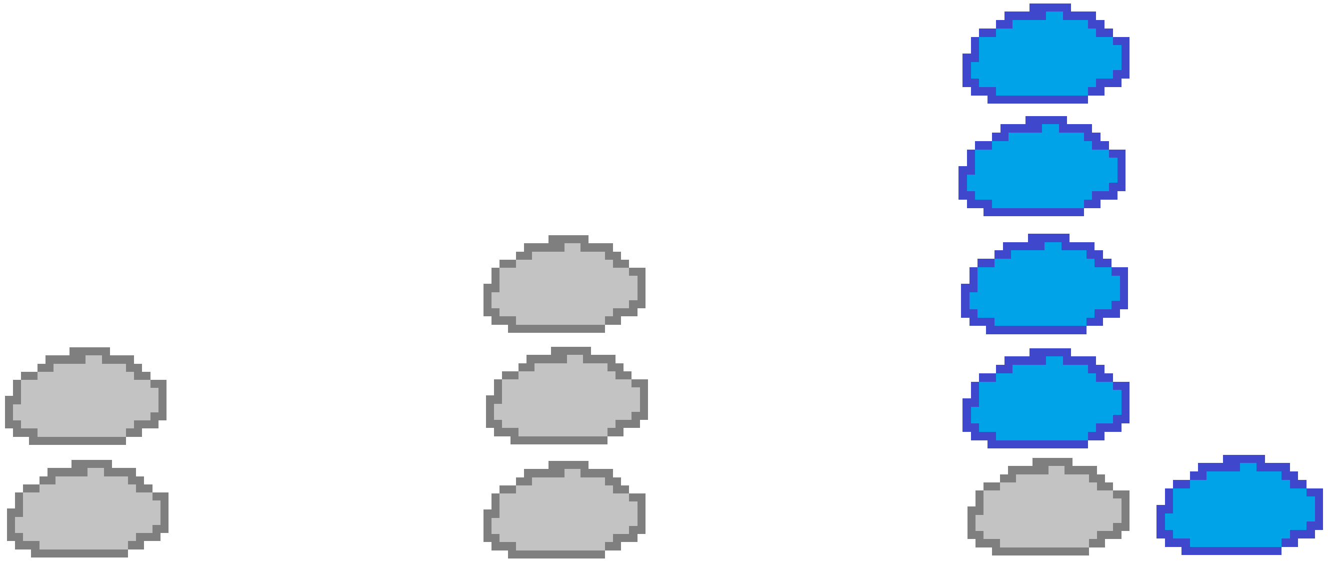


# Sample play

Player 1: 

Player 2: 

- $N = 3, a_1 = 5, a_2 = 3, a_3 = 8$

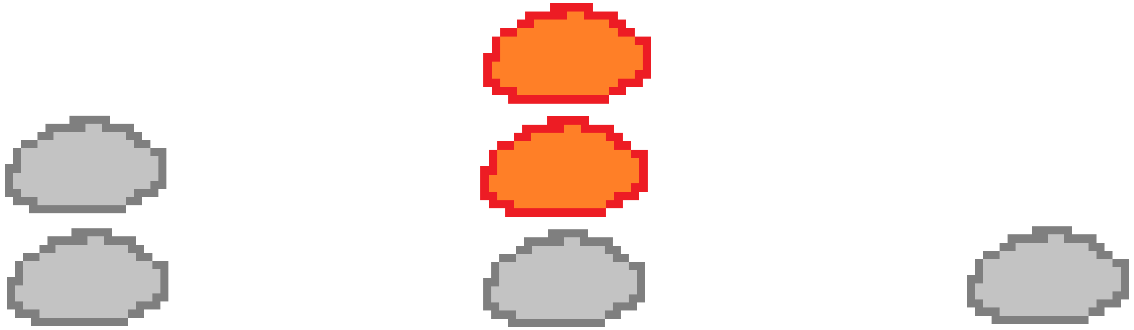


# Sample play

Player 1: 

Player 2: 

- $N = 3,\ a_1 = 5,\ a_2 = 3,\ a_3 = 8$

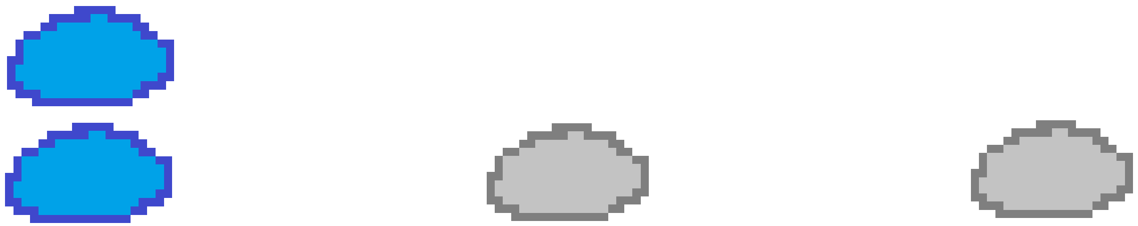


# Sample play

Player 1:

Player 2:

- $N = 3$, $a_1 = 5$, $a_2 = 3$, $a_3 = 8$



# Sample play

Player 1: 

Player 2: 

- $N = 3,\ a_1 = 5,\ a_2 = 3,\ a_3 = 8$

# Sample play

Player 1: 

Player 2: 

- $N = 3$, $a_1 = 5$, $a_2 = 3$, $a_3 = 8$

Player 1: 

Player 2: 

# Sample play

- $N = 3$, $a_1 = 5$, $a_2 = 3$, $a_3 = 8$

(To slide 61)

# Does Player 2 have a chance?

- No, not if player 1 is good
- In this particular example, player 1 wins under perfect play

# When N is small

- If N = 1, Player 1 wins

- If N = 2,
  - a1 = a2: Player 2 wins (copy what Player 1 does)
  - a1 != a2: Player 1 wins (make two piles the same)

- This is not even close to the general solution though...

# General solution

- Let $\oplus$ be the bitwise xor operator
  - $(0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0)$

```
If a_1⊕a_2⊕...⊕a_N = 0, player 2 wins.
Otherwise, player 1 wins.
```

If $a_1 \oplus a_2 \oplus \ldots \oplus a_N = 0$, player 2 wins.
Otherwise, player 1 wins.

- Because bitwise xor is helpful in analysing Nim, we also call it the **Nim-sum**

# Winning Strategy

- If $a_1 \oplus a_2 \oplus ... \oplus a_N = 0$,
  - After Player 1's move, $a_1 \oplus a_2 \oplus ... \oplus a_N$ != 0
  - Important fact: there exists a move for Player 2 so that after the move $a_1 \oplus a_2 \oplus ... \oplus a_N = 0$ (Proof next slide)
- If $a_1 \oplus a_2 \oplus ... \oplus a_N$ != 0
  - Player 1 should move to make $a_1 \oplus a_2 \oplus ... \oplus a_N = 0$
  - Then, use the strategy above

# Why does it work?

- If Nim-sum is not zero,

```
     xxxxxxxxxx
     xxxx1xxxxx
     xxxxxxxxxx
⊕    xxxxxxxxxx
  --------------
     00001xxxxx
```

- Find the most significant 1
- There exists a number with ‘1’ in that place
- Reduce that number so that xor-sum is zero

# Example

- 5, 3, 8

```
      0101
      0011
 ⊕    1000
 --------
      1110
```

- Find the most significant 1

# Example

- 5, 3, 8

```
    0101
    0011
⊕   1000 **
--------
    1110
```

- Find the most significant 1
- There exists a number with ‘1’ in that place

# Example

- 5, 3, 8

```
     0101
     0011
⊕    1000 --> 0110
--------
     1110
```

- Find the most significant 1
- There exists a number with ‘1’ in that place
- Reduce that number so that xor-sum is zero

# Another example

- 12, 10, 5

```
    1100
    1010
⊕   0101
--------
    0011
```

- Find the most significant 1

# Another example

- 12, 10, 5

```
    1100
    1010 **
⊕   0101
--------
    0011
```

- Find the most significant 1
- There exists a number with ‘1’ in that place

# Another example

- 12, 10, 5

1100

10**10** --> 10**01**

$\oplus$ 0101

--------

00**11**

- Find the most significant 1
- There exists a number with ‘1’ in that place
- Reduce that number so that xor-sum is zero

- Let’s watch sample play again :)
- Can you find the winning strategy at each step?

# Mini-test

- Nim game
- Suppose N = 4, $a_1$ = 35, $a_2$ = 18, $a_3$ = 27

- Now, Player 1 is kind (and stupid) enough to let Player 2 decide $a_4$

- In order to win, what should Player 2 choose for $a_4$?

# Answer

- $a_4 = 42$

- Player 2 wants $a_1 \oplus a_2 \oplus a_3 \oplus a_4 = 0$
- Hence, $a_1 \oplus a_2 \oplus a_3 = a_4$

- $a_4 = 35 \oplus 18 \oplus 27 = 42$

- Note that the answer is **unique**

# Blank slide

- Rest / Q&A

# Why discuss Nim?

- Nim is considered to be the *prototypical* game among the impartial combinatorial games

- Other games can be analyzed using the idea of Nim!

# Introducing another idea...

- Sprague-Grundy function, or simply SG funciton
- Development of the idea of P-/N-positions

# MEX (**M**inimal **EX**cludant)

- mex of a set of non-negative integers is the smallest non-negative integer *not in the set*

- Examples:
  - mex({0, 2, 4}) = 1
  - mex({0, 5, 1, 3}) = 2
  - mex({2, 0, 1}) = 3
  - mex({1, 3, 5}) = 0
  - mex({all positive integers}) = 0
  - mex(empty set) = 0

# Back to SG function

- Suppose we can reach positions $x_1$, $x_2$, $x_3$, ..., $x_k$ from the current position $X_{cur}$

$$
\begin{aligned}
SG(X_{cur}) &= mex(\{SG(x_i)\}) \\
&= mex(\{SG(x_1), SG(x_2), ..., SG(x_k)\})
\end{aligned}
$$

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | | | | | | | | | | | |

0 is an ending position

>> SG(0) = SG({}) = 0

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | | | | | | | | | | |

$$SG(1) = mex(\{0\}) = 1$$

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | | | | | | | | | |

SG(2) = mex({0, 1})
= 2

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | | | | | | | | |

SG(3) = mex({0, 1, 2})
= 3

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | 0 | | | | | | | |

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

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | 0 | 1 | | | | | | |

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

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | 0 | 1 | 2 | | | | | |

SG(6) = mex({3, 0, 1})
= 2

• • • • • •

# Example

- Take-away game with K = 3

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |

# Comparing SG value and P-/N-positions

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SG(N) | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P/N? | P | N | N | N | P | N | N | N | P | N | N | N |

# Basic Properties of SG function

- A position is a P-position iff its SG value = 0
- A position is an N-position iff its SG value > 0

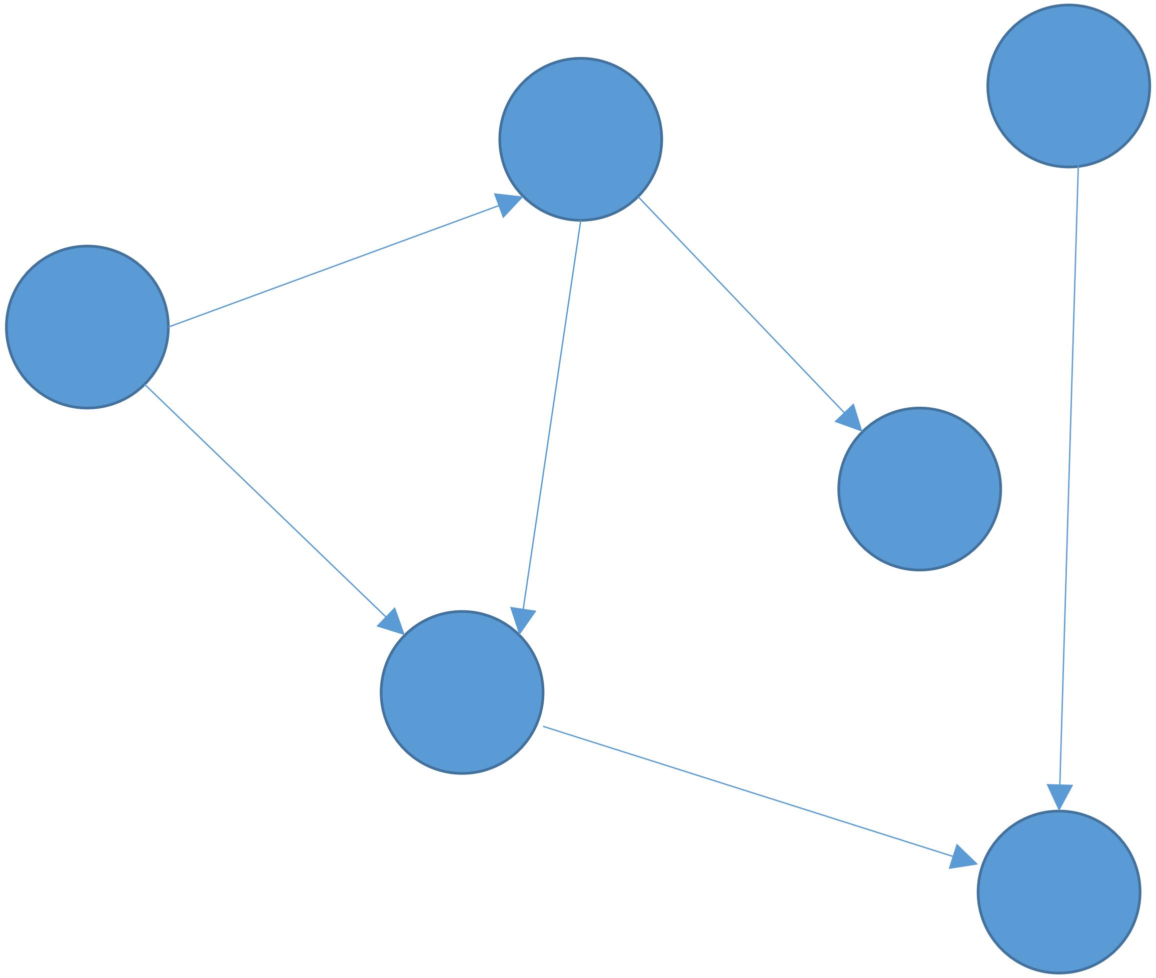
- An ending position has SG value = 0

- Let initial position be $P_0$
  - If $SG(P_0) > 0$, Player 1 wins (N-position)
  - If $SG(P_0) = 0$, Player 2 wins (P-position)

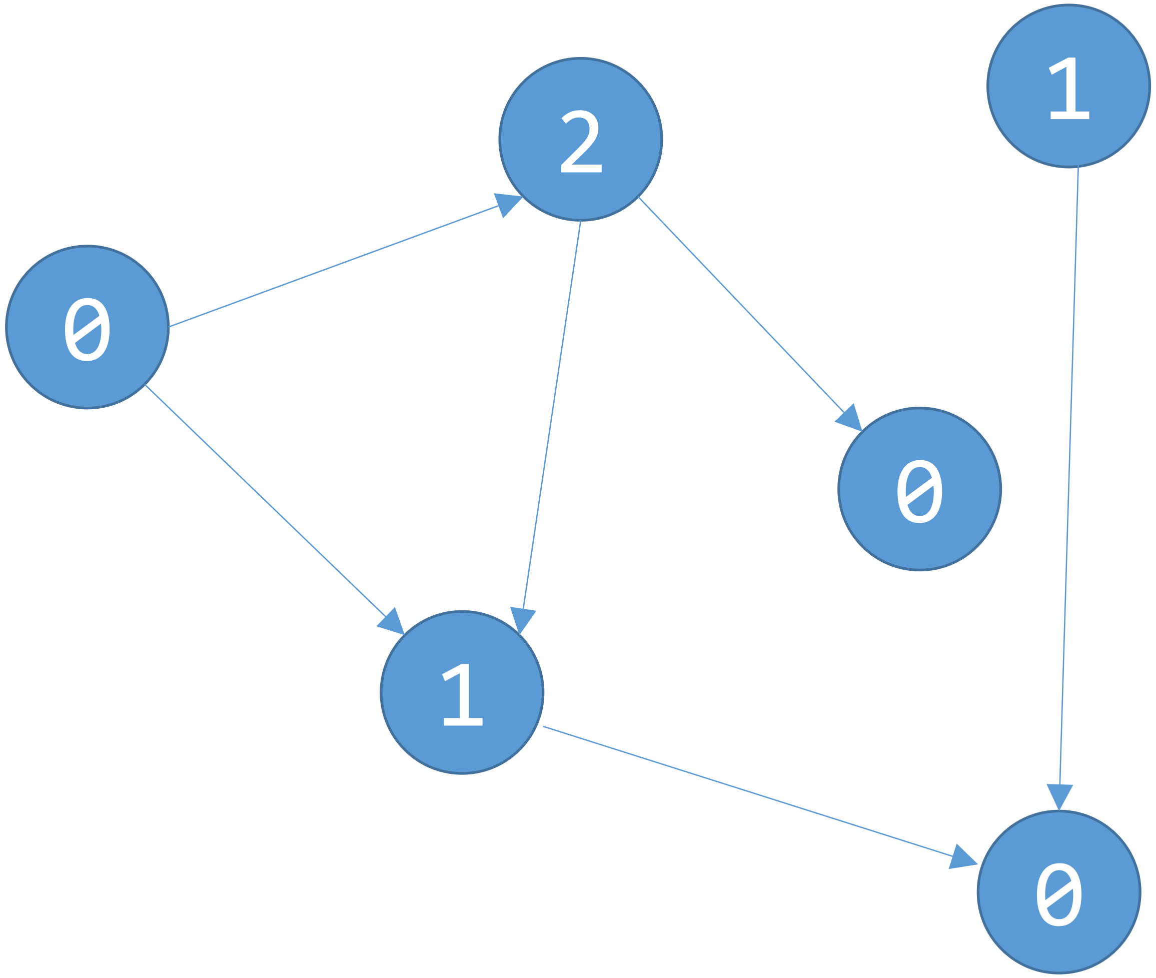
# Mini-test

- Game represented by a DAG (directed acyclic graph)
- Node = game state, edge = move
- A --> B means one can move from state A to state B
- Fill in the SG values of each state (node)

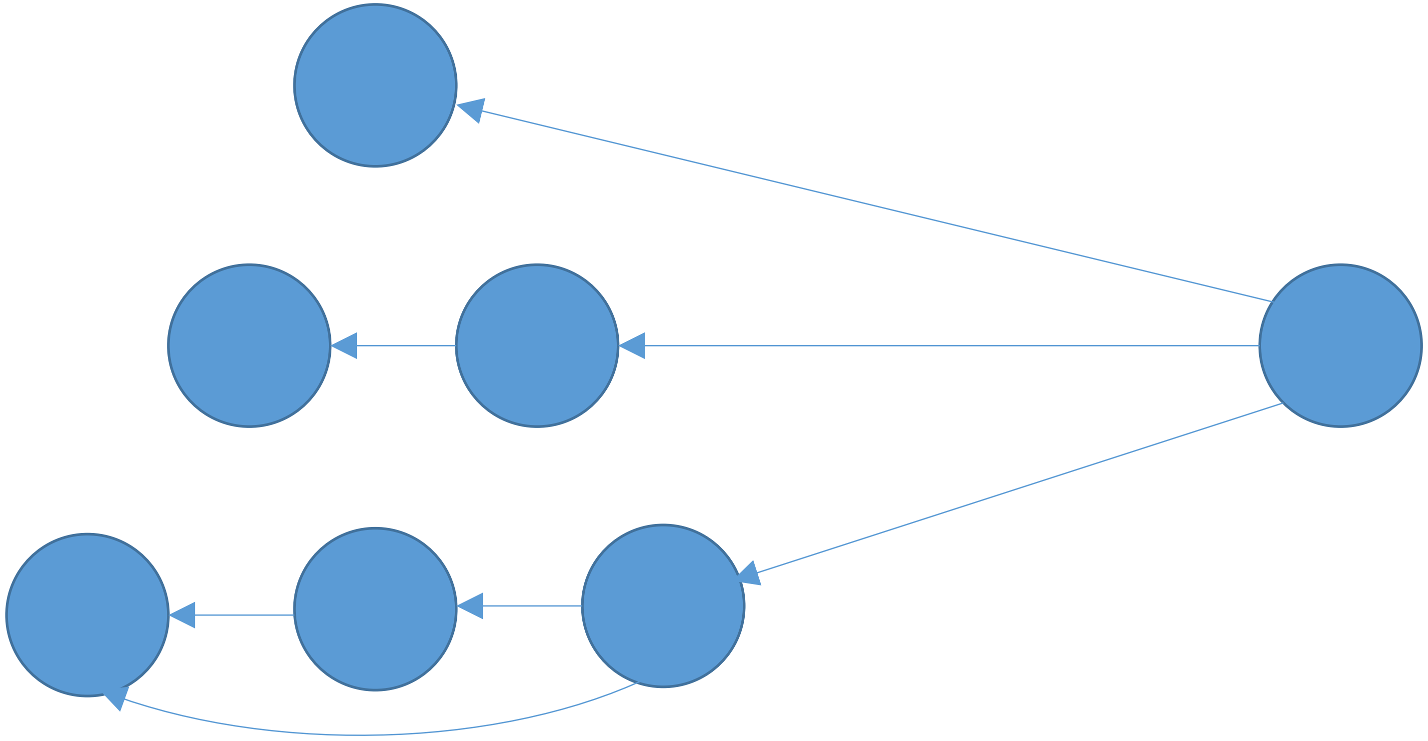
# Graph 1



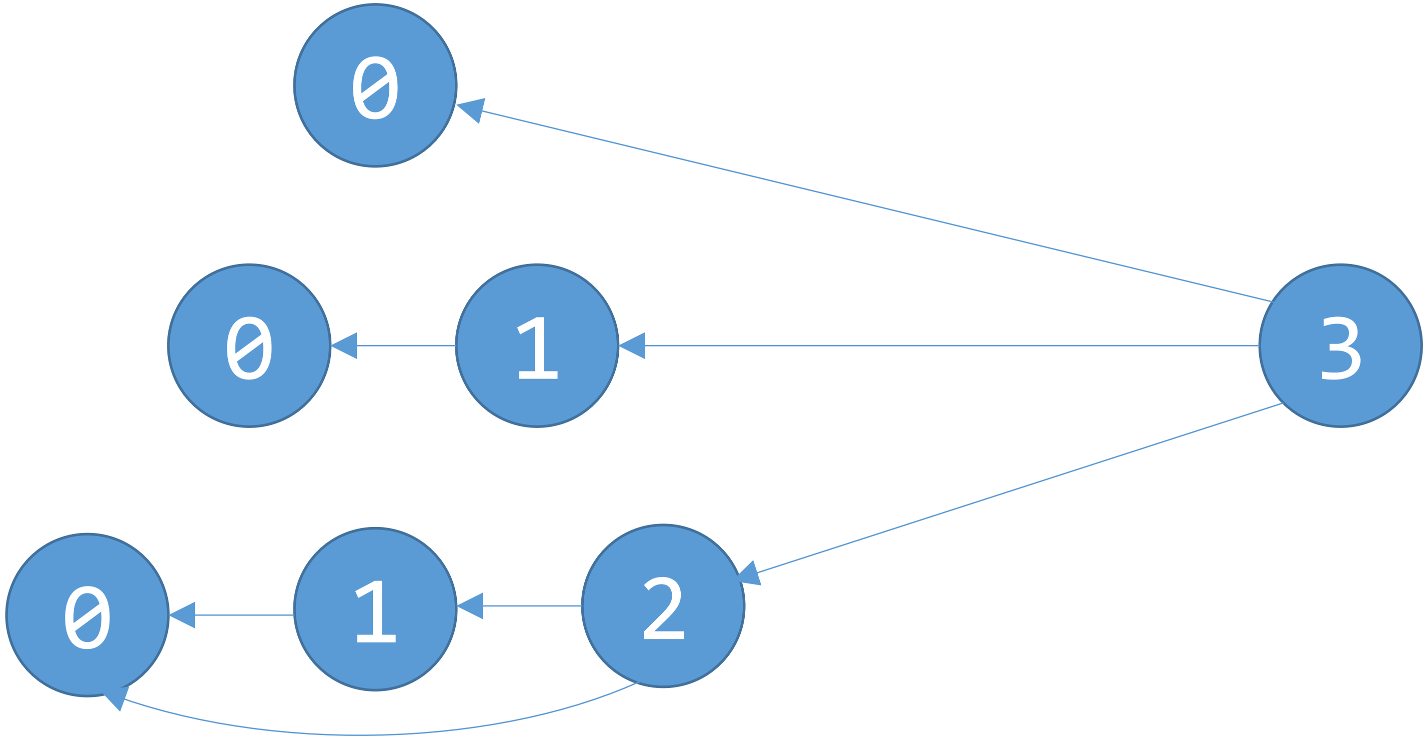
# Solution



# Graph 2



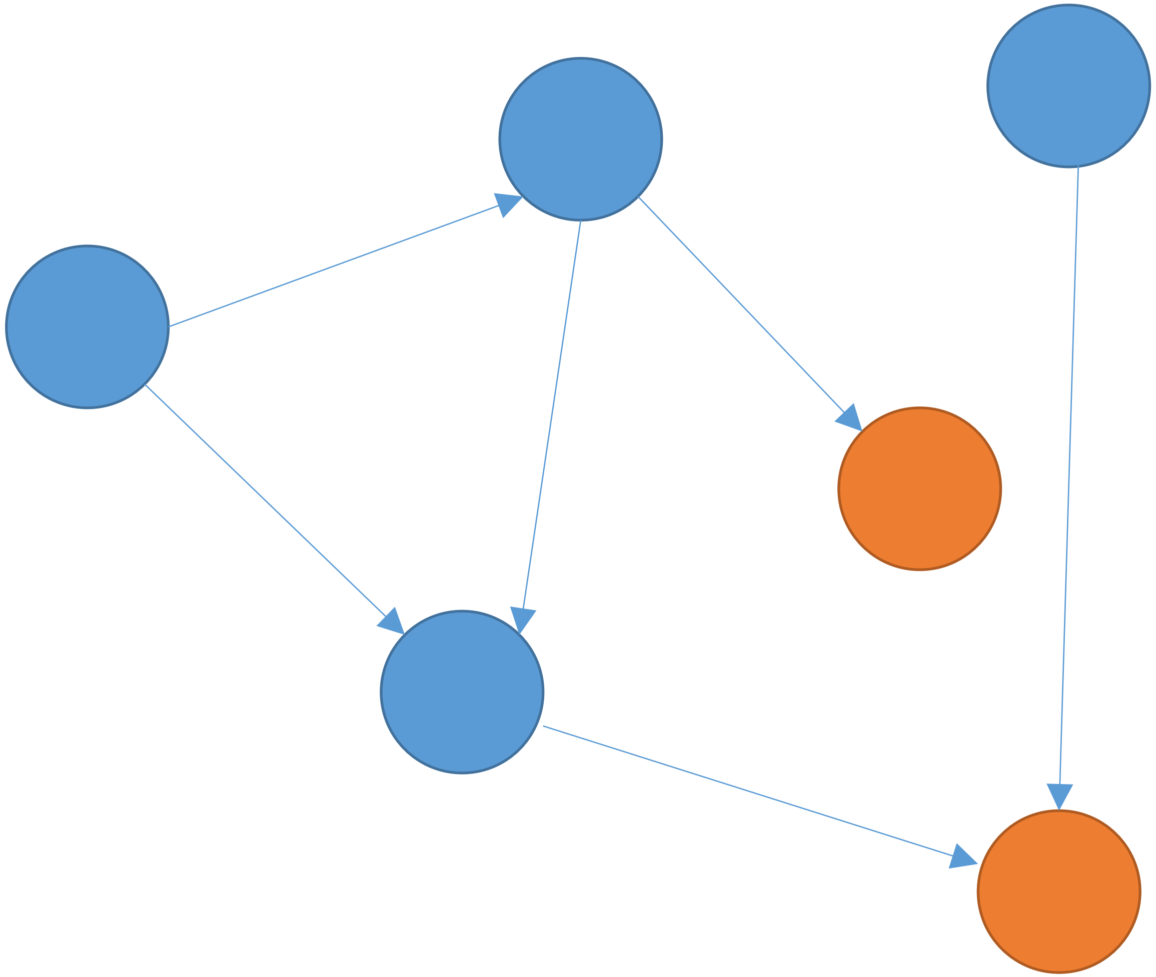
# Solution



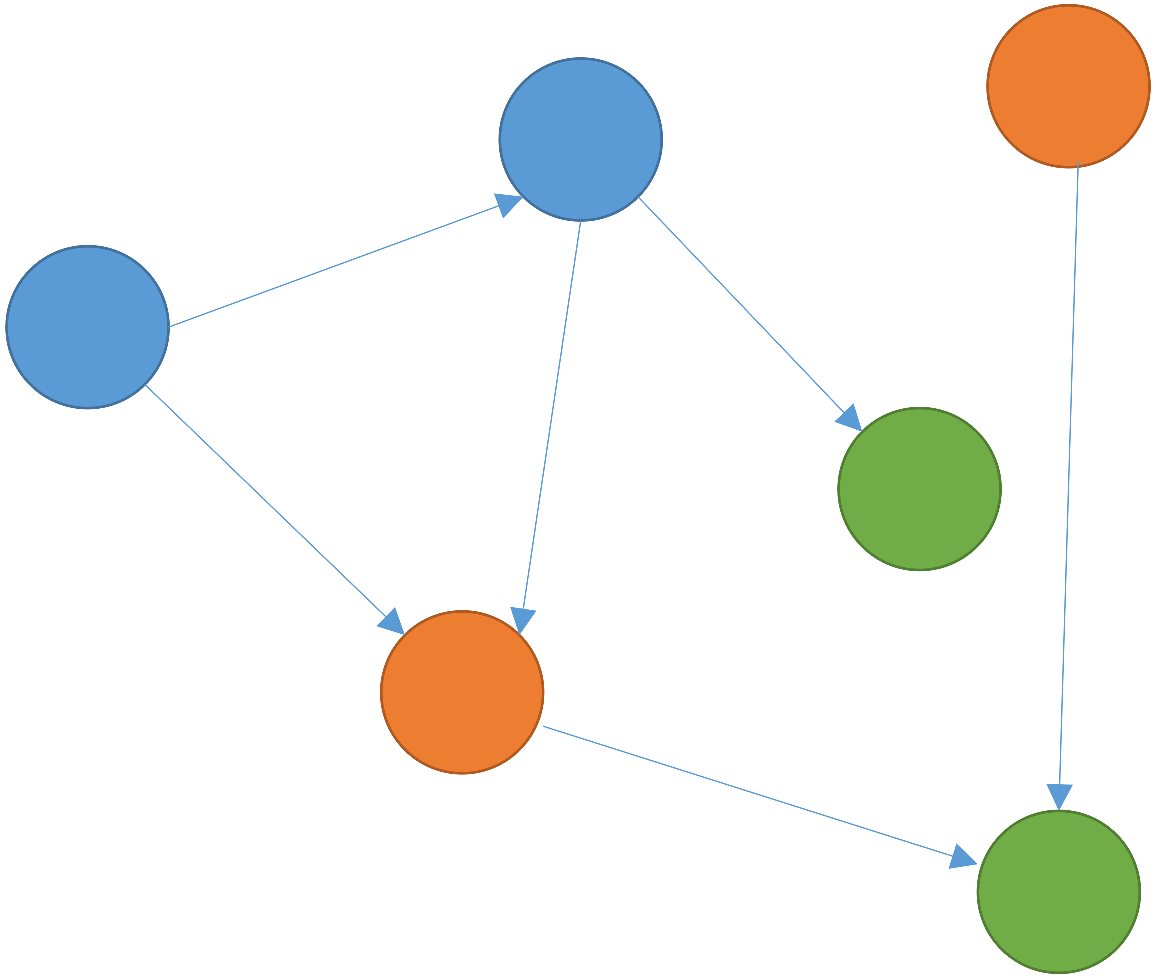
# Order of filling

- Any topological ordering
- Find one using topological sort
- In most combinatorial games, a topological ordering may be as simple as 1, 2, 3, ...

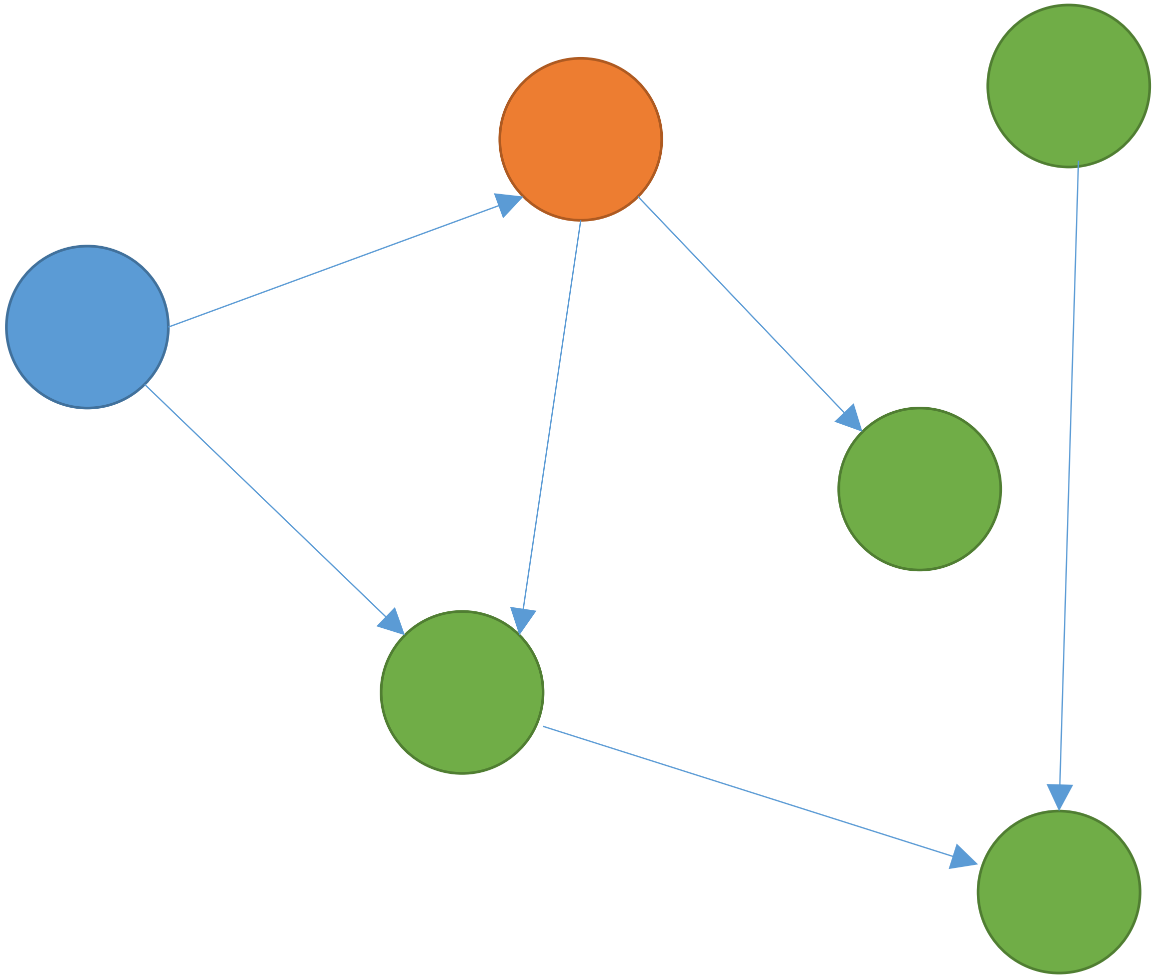
# Graph 1 ordering



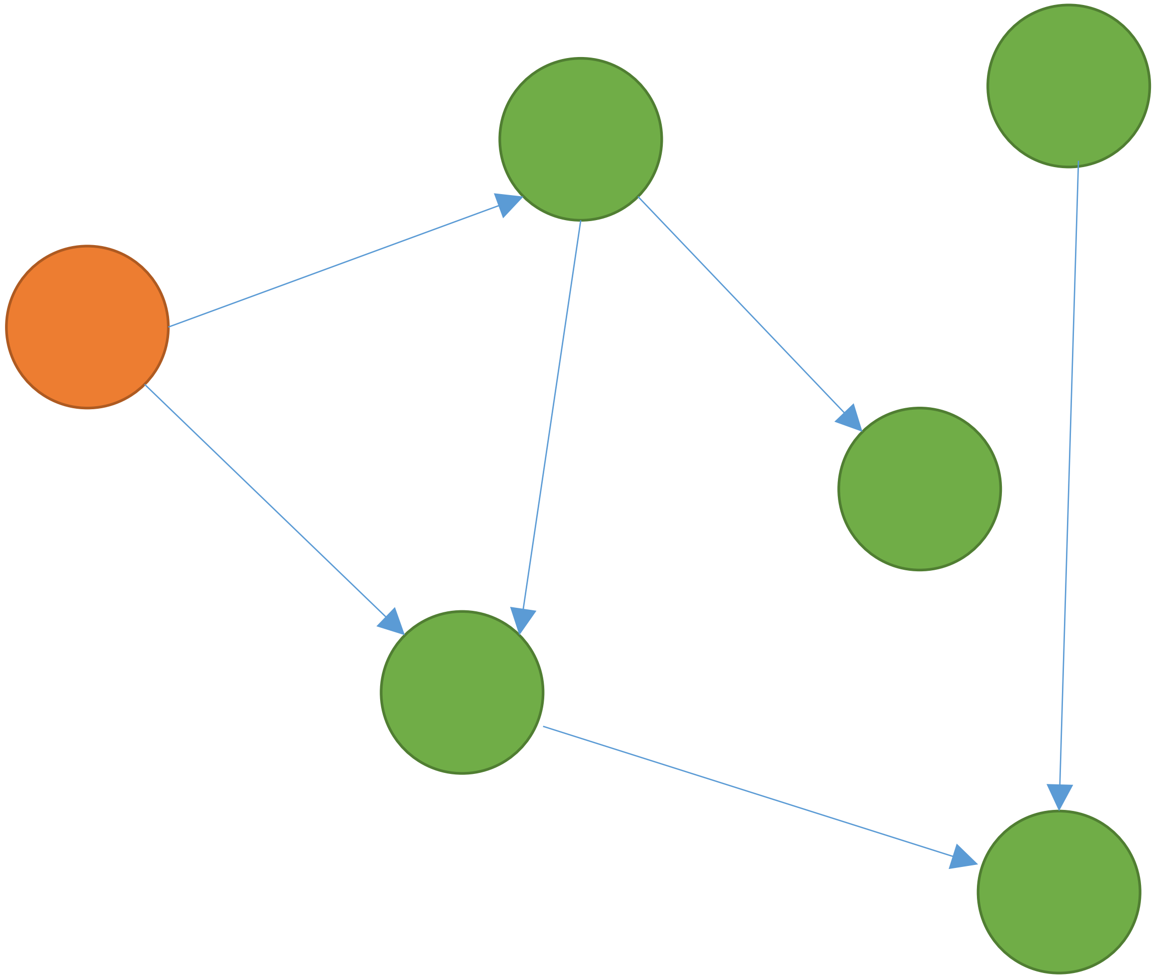
# Graph 1 ordering



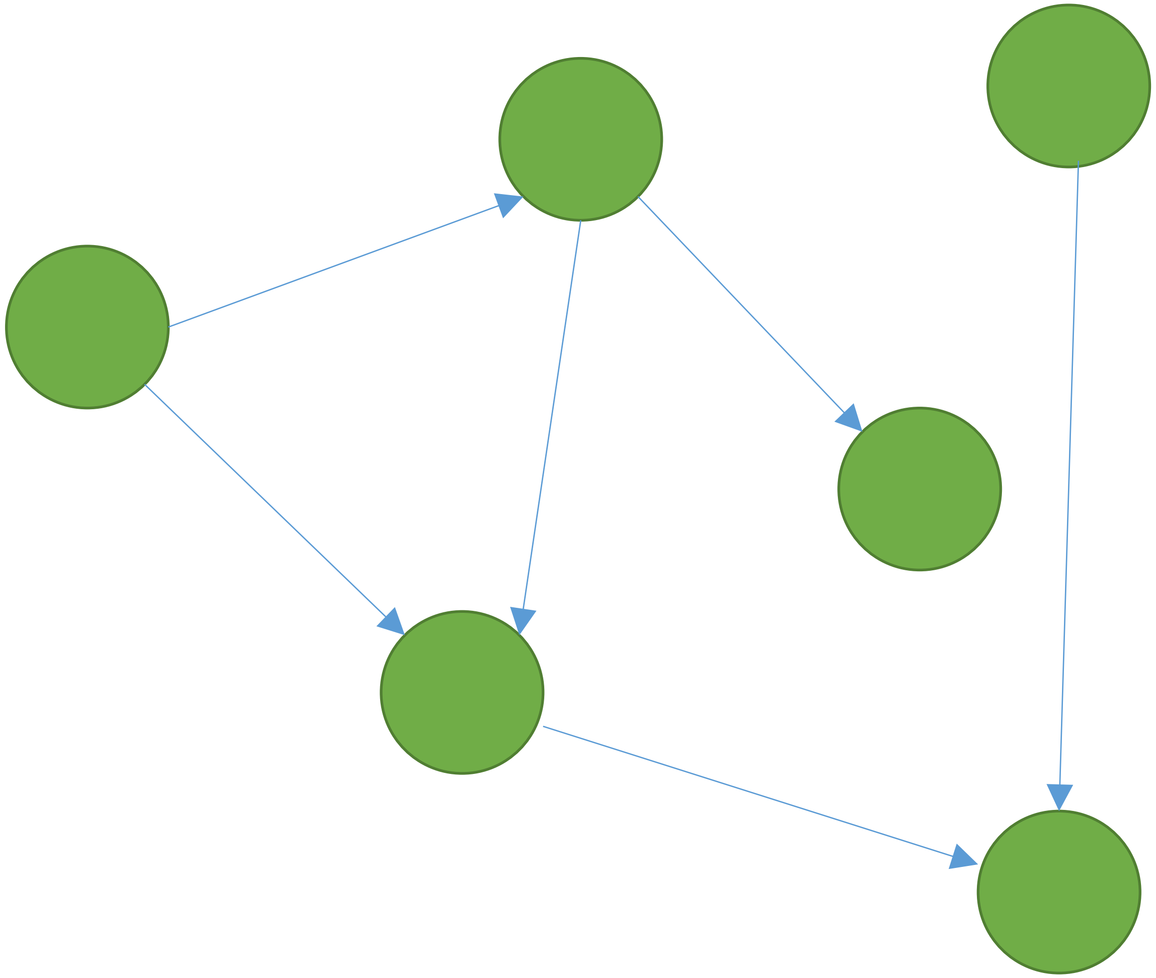
# Graph 1 ordering



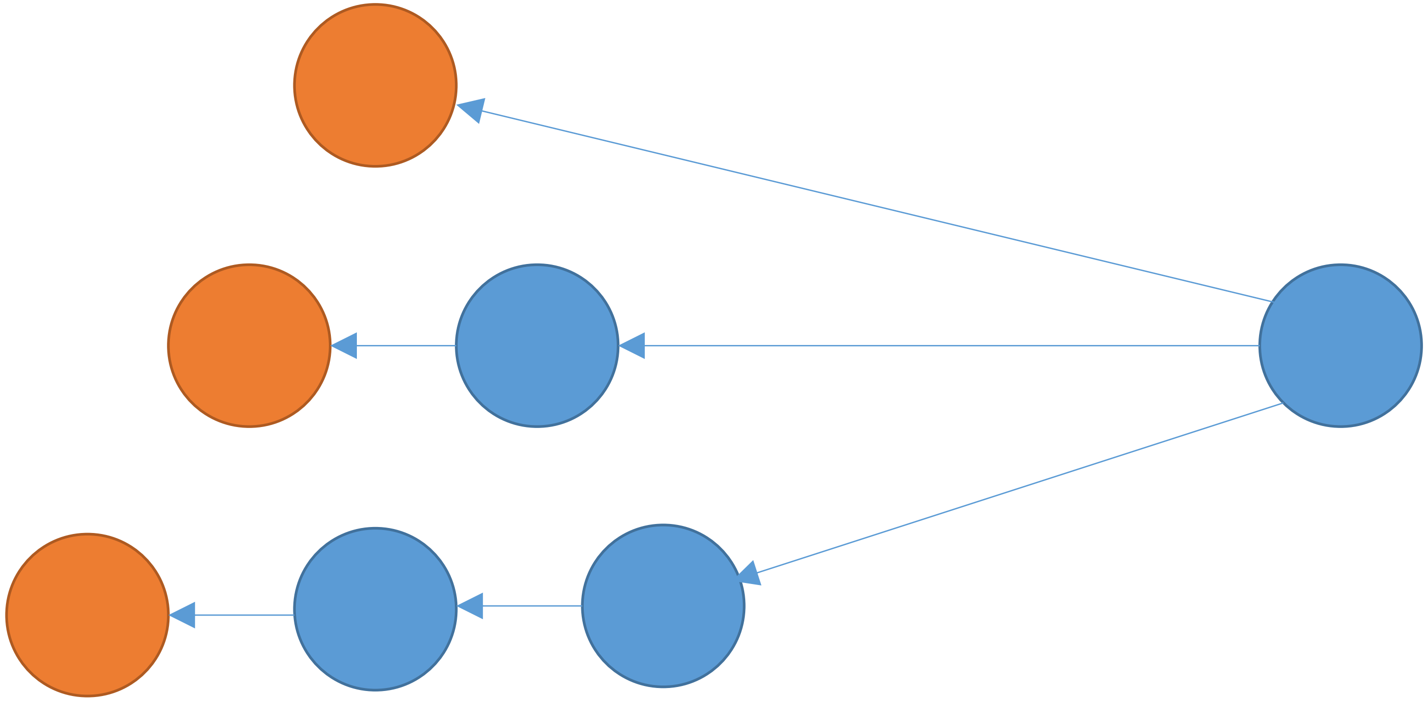
# Graph 1 ordering



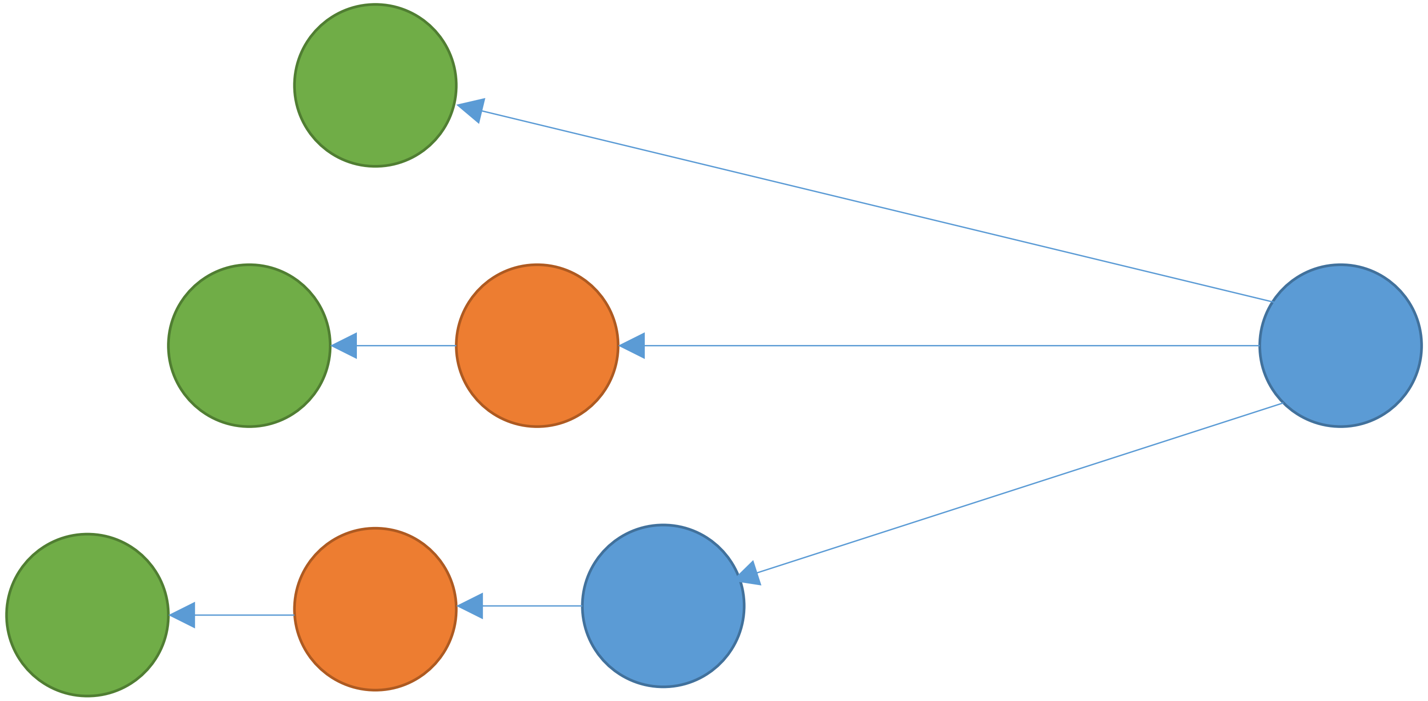
# Graph 1 ordering



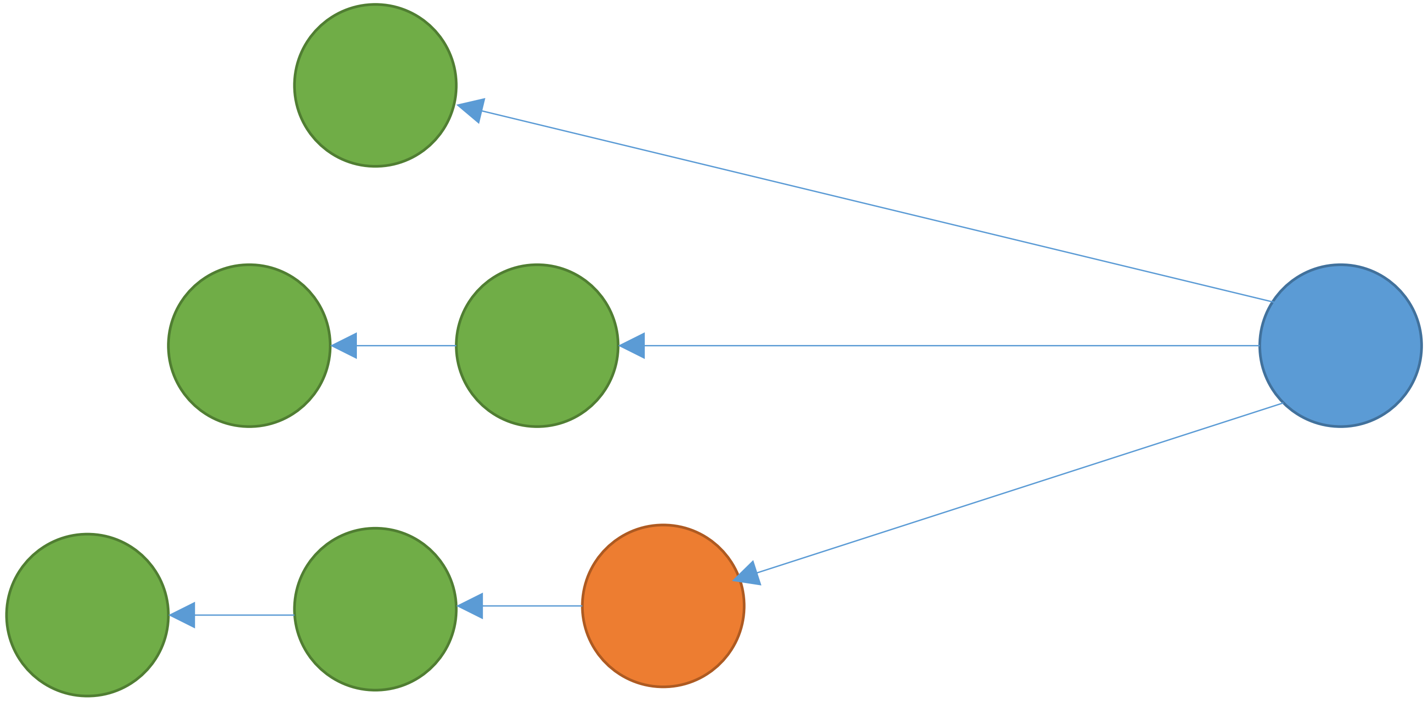
# Graph 2 ordering



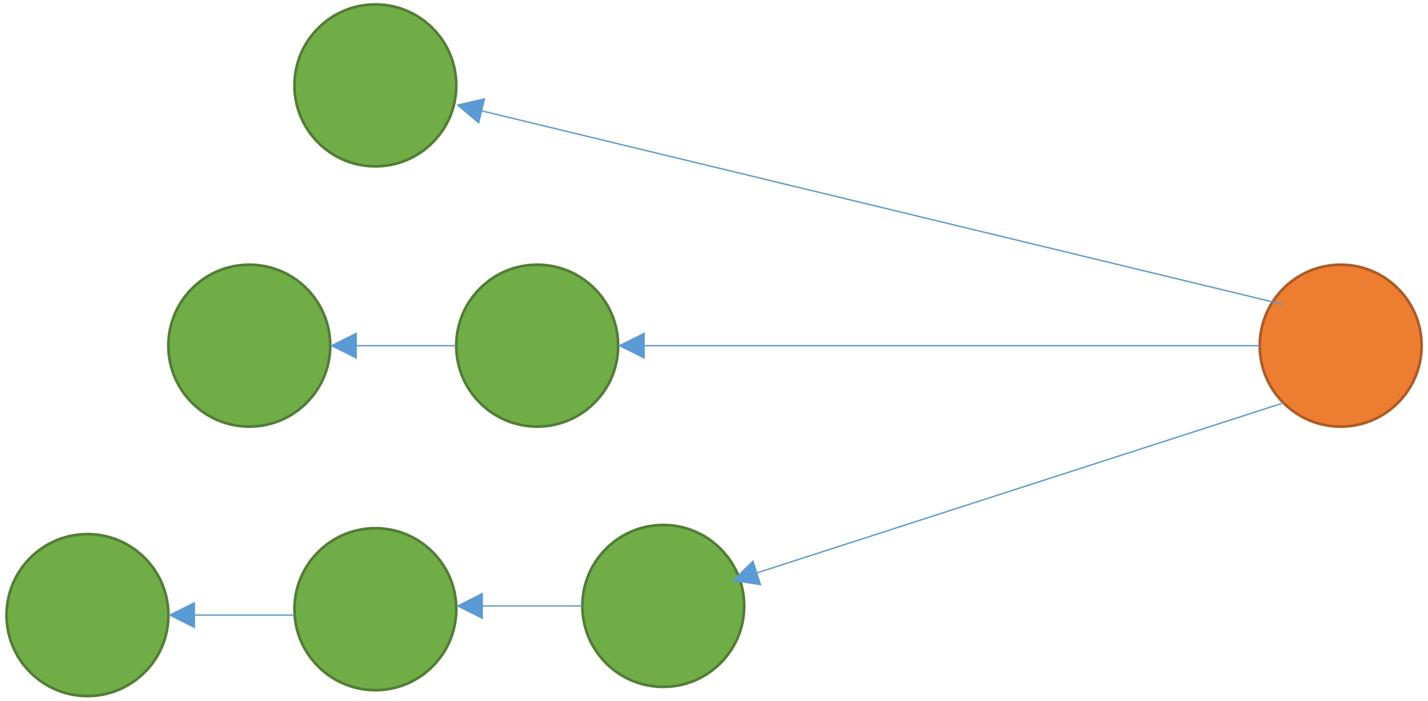
# Graph 2 ordering



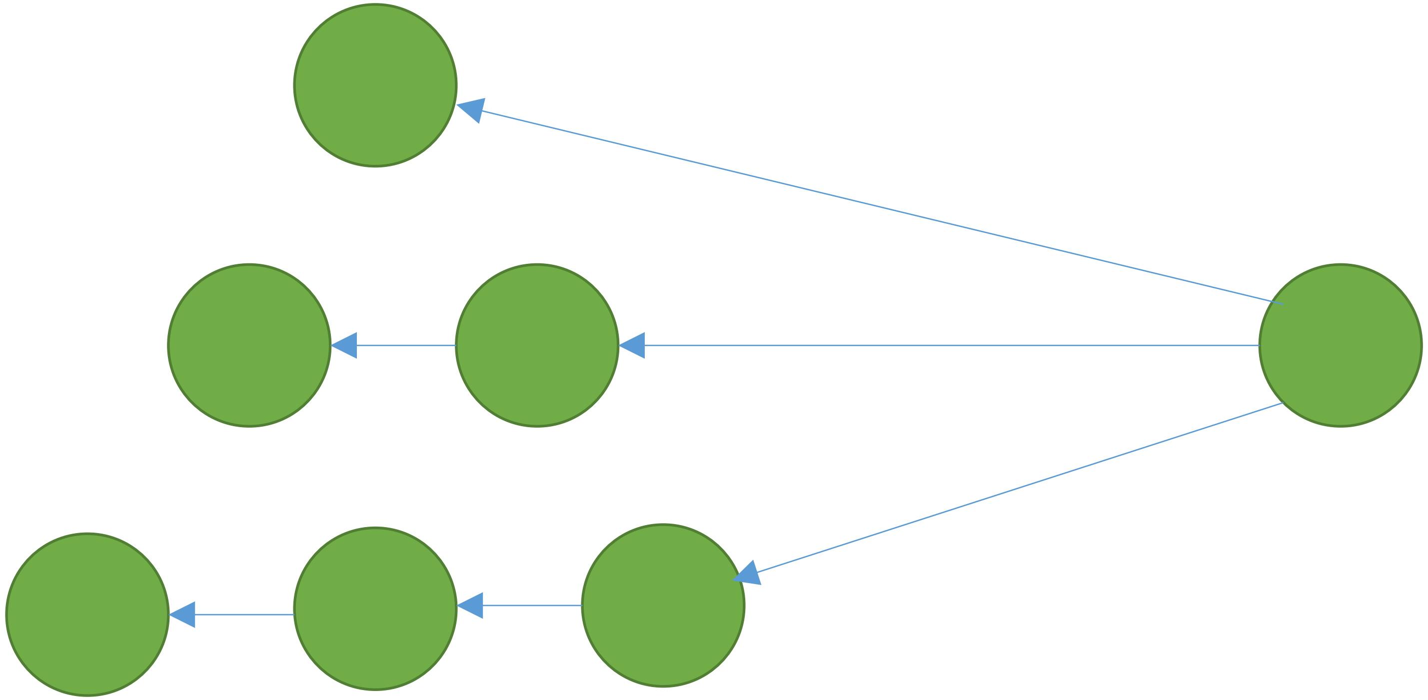
# Graph 2 ordering



# Graph 2 ordering



# Graph 2 ordering



# Blank slide

- Rest / Q&A

# Nim again, using SG function

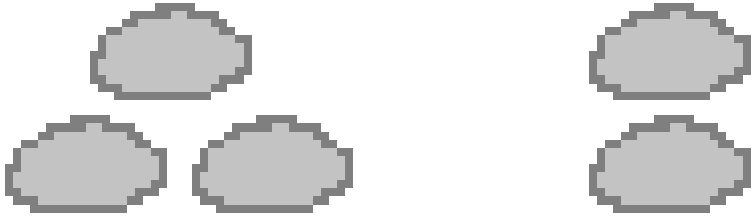
- One pile (N = 1)
- Let $G_K$ be the game state with K stones

- We can reach $G_0$, $G_1$, ..., $G_{K-1}$ from $G_K$
- $SG(G_K) = mex(\{SG(G_0), SG(G_1), ..., SG(G_{K-1})\})$

- We can show that $SG(G_K) = K$ for all K

# Nim again, using SG function

- Two piles (N = 2)
- Let $G_{X,Y}$ be the game state with X stones in one pile and Y stones in another
- e.g. $G_{3,2}$ =



- We can reach, from $G_{X,Y}$, the following positions:
  - $G_{0,Y}$, $G_{1,Y}$, ..., $G_{X-1,Y}$
  - $G_{X,0}$, $G_{X,1}$, ..., $G_{X,Y-1}$

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | | | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | 2 | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | 2 | 5 | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | 2 | 5 | 4 |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

• • • • • •

# SG values of $G_{X,Y}$

| $SG(G_{X,Y})$ | X = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Y = 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 0 | 3 | 2 | 5 | 4 |
| 2 | 2 | 3 | 0 | 1 | 6 | 7 |
| 3 | 3 | 2 | 1 | 0 | 7 | 6 |
| 4 | 4 | 5 | 6 | 7 | 0 | 1 |
| 5 | 5 | 4 | 7 | 6 | 1 | 0 |

# Well...

- You have a feeling that relying on SG function alone would make things difficult when N is larger
- We know $SG(G_K) = K$, which is quite a simple function
- Maybe $SG(G_{X,Y})$ relies on $SG(G_X)$ and $SG(G_Y)$ in some way?

# Of course...

- $SG(G_{X,Y}) = SG(G_X) \oplus SG(G_Y)$
- In other words, $SG(G_{X,Y}) = X \oplus Y$

# Generalising to all N

- Let $G_{a1,a2,...,aN}$ represent the state of a Nim game with N (possibly empty) piles of stones and $a_i$ stones in the i-th pile

- $SG(G_{a1,a2,...,aN}) = SG(G_{a1}) \oplus SG(G_{a2}) \oplus ... \oplus SG(G_{aN})$

- Which means $SG(G_{a1,a2,...,aN}) = a_1 \oplus a_2 \oplus ... \oplus a_N$, exactly the **Nim-sum**!

# Why it is true that $SG(G_{a1,a2,...,aN}) = a_1 \oplus a_2 \oplus ... \oplus a_N$

- By induction (on $S = a_1 + a_2 + ... + a_N$)

- Let $G_{cur} = G_{a1,a2,...,aN}$ be the current state
- If Nim-sum of $G_{cur} = 0$, $G_{cur}$ is a P-position, so $SG(G_{cur}) = 0$

- If Nim-sum of $G_{cur} > 0$, for any $K <$ $G_{cur}$, *one can always reduce one pile to reach $G_{new}$ so that Nim-sum of $G_{new} = K$*. But Nim-sum of $G_{new} = SG(G_{new})$ by induction hypothesis. So, $SG(G_{cur}) >=$ Nim-sum of $G_{cur}$
- For any $G_{new}$, $SG(G_{new})$ != Nim-sum of $G_{cur}$
- Hence, $SG(G_{cur})$ = Nim-sum of $G_{cur}$

(Verify the *underlined* part by yourself)

# Why is Nim closely related to SG?

- Because of the following crucial property:

```
Let SG(P) = K.
  - If K > 0, you can reach a position Q with
    SG(Q) = K’ for any K’ < K
  - If K = 0, you either cannot move or have
    to reach a position Q with SG(Q) > 0
```

- So similar to the rules of (one-pile) Nim!

# Sprague-Grundy Theorem

- Suppose we have N **independent** games: $G_1$, $G_2$, ..., $G_N$
- Each turn, a player can pick one game and make a valid move in that game
- Same ending condition: the player with no moves loses

# Sprague-Grundy Theorem

- Suppose we have analysed each independent game using SG function
- Let $P_i$ be the position in $G_i$

Then, $SG(P_1, P_2, ..., P_N)$
$= SG(P_1) \oplus SG(P_2) \oplus ... \oplus SG(P_N)$

# Proof?

- Using the strategy for playing Nim and the fact that $SG(G_{a1,a2,...,aN}) = a_1 \oplus a_2 \oplus ... \oplus a_N$

# Implication

- Because of the relationship between SG function and Nim, every game can be played in the way of Nim

- N independent games = N-pile Nim!

- We do not need to analyse (i.e. find the SG values of) N-dimensional positions to solve the compound game

# End of theory part

- Next up: problem-solving session

- Useful reference: https://www.math.ucla.edu/~tom/Game_Theory/Contents.html

- So far, we have covered a portion of Part I

# Blank slide

- Rest / Q&A

# Problem-solving session

- 9 problems

- Solve = Describe an efficient algorithm to find the answer. Not necessarily something ‘mathematical’ or elegant.

# Strategy

- Look for invariants
- Try small cases
- Analyse the game using SG function
- Sum of games can be dealt with by Nim-sum ($\oplus$)
  - So, no need to worry about "N piles of stones"

- If you want to convince yourself, prove your results using induction

# Problem 1

(CF 705B)

- N (<= 100000) piles of stones
- i-th pile has $a_i$ (<= $10^9$) stones
- Each move = split a pile of X stones into two piles of Y and Z stones, Y > 0, Z > 0, X = Y + Z

- For each K <= N, determine who wins if the game is played with the first K piles

# Problem 2

(Codechef June CHCOINSG)

- One pile of N (<= $10^9$) stones
- Each move = remove $p^k$ stones, where p is prime and k is a **non-negative** integer

- Determine who wins

# Problem 3

(CF 15C)

- N **groups** of stones piles
- Given $K_i$, $M_i$, the i-th group has $M_i$ piles of stones, with $K_i$, $(K_i+1)$, $(K_i+2)$, ..., $(K_i+M_i-1)$ stones
- Use all $\Sigma M_i$ stone piles to play Nim
- Determine who wins
- K, M <= $10^{16}$

# Problem 4

(ACM ICPC Jakarta 2010)

- N (<= 100) piles of stones
- i-th pile has $a_i$ (<= $2*10^{18}$) stones
- Each move = choose a pile, remove at least one but no more than $a_i/2$ stones

- Determine who (Player 1 or 2) wins

# Problem 5

(Hackerrank "5 days of game theory")

- N (<= 100) integers $a_1$, ..., $a_N$ (<= $10^5$)
- Each move = pick one integer X, delete it, and write Y (> 1) copies of integer Z, where X = YZ

- As usual, the player who cannot move loses
- Determine who wins

# Problem 6

(CF 335C)

- 2 x N grid (N <= 100), with some obstacles
- Each move = place an obstacle in an empty cell, *without blocking the flow from left to right*
- i.e. There should be a path from left to right after each move

- As usual, the player who cannot move loses
- Determine who wins

# Problem 7

(AtCoder Grand Contest 010 D)

- N (<= $10^5$) positive integers $a_1$, $a_2$, ... $a_N$ (<= $10^9$)
- gcd($a_1$, $a_2$, ..., $a_N$) = 1
- Each move = replace some $a_i$ ($a_i$ > 1) by $a_i$ - 1
- Afterwards, divide each $a_i$ by gcd($a_1$, $a_2$, ..., $a_N$)

- As usual, the player who cannot move loses
- Determine who wins

# Problem 8

(CF 167C)

- Two integers A and B ($0 \le A, B \le 10^{18}$)
- Two types of moves (let A <= B):
  - (A, B) -> (A, $B - A^k$), k > 0, $B - A^k \ge 0$
  - (A, B) -> (A, B mod A)

- As usual, the player who cannot move loses
- Determine who wins

# Problem 9

(CF 39E)

- Given A, B, N (A <= 10000, B <= 30, N <= $10^9$)
- Initially, $A^B < N$

- Each move = increase A or B by 1
- After each move, $A^B < N$ must hold

- As usual, the player who cannot move loses
- Determine who wins (Note: a draw is possible)

# Solution session

# The End