

KYLE DAVIS

# ALGORITHMS

PUBLISHER



# Contents

|                               |    |
|-------------------------------|----|
| <i>Chapter 1: Lists</i>       | 5  |
| <i>Chapter 2: Trees</i>       | 7  |
| <i>Chapter 3: Graphs</i>      | 9  |
| <i>Chapter 4: Game Theory</i> | 11 |
| <i>Index</i>                  | 15 |



# *Chapter 1: Lists*

*Array Lists*

*Linked Lists*



## *Chapter 2: Trees*





## *Chapter 3: Graphs*



## Chapter 4: Game Theory

### *The Game of Nim*

Nim is a very simple game that can be used to demonstrate fundamental theorems in game theory. In the game of Nim there are piles of stones and two players. Each player alternates taking turns, the first player that cannot take a turn loses. In each turn a player takes some positive number of stones from any pile with a sufficient number of stones left. So a player loses when all of the piles are empty and it is their turn.

The starting configuration of Nim is different from game to game. For example a game could start with 3 piles each containing 5 stones, or a game could start with 3 piles of containing 4, 5, and 6 stones. The question is: given a starting configuration who will win if they play optimally?

To determine the winner of a particular game of Nim there is a very simple algorithm. Take the xor of the number of stones in each pile, if this value is 0 the second player will win, otherwise the first player will win. This solution is not intuitive, so let's explore why it is true by working through an example.

### *Example*

Consider a game where there are 5 piles with 3, 6, 1, 2, and 7 stones. First let's compute the xor of all piles. (The right column is each number's binary representation)

|   |     |
|---|-----|
| 3 | 011 |
| 6 | 110 |
| 1 | 001 |
| 2 | 010 |
| 7 | 111 |
| 1 | 001 |

According to our rules above, in this example the player that goes first (we shall refer to her as Alice, and Bob) will win. Why is that? Let's examine a possible sequence of moves between Alice and Bob.

|       |   |
|-------|---|
| Start | $011 \oplus 110 \oplus 001 \oplus 010 \oplus 111 = 001$ |
| Alice | $011 \oplus 110 \oplus 000 \oplus 010 \oplus 111 = 000$ |
| Bob   | $000 \oplus 110 \oplus 000 \oplus 010 \oplus 111 = 011$ |
| Alice | $000 \oplus 110 \oplus 000 \oplus 001 \oplus 111 = 000$ |
| Bob   | $000 \oplus 000 \oplus 000 \oplus 001 \oplus 111 = 110$ |
| Alice | $000 \oplus 000 \oplus 000 \oplus 001 \oplus 001 = 000$ |
| Bob   | $000 \oplus 000 \oplus 000 \oplus 000 \oplus 001 = 001$ |
| Alice | $000 \oplus 000 \oplus 000 \oplus 000 \oplus 000 = 000$ |

Alice wins

Looking at each of Alice's moves, she always makes a move the changes the xor of the game to be 0. This means that every time Bob makes a move he must change the xor of the game to be non-zero. It's impossible for it to remain zero after Bob's turn because he can only change one pile, and each bit in that pile that changes will flip a bit in the xor of the entire game. Since Bob must take some number of stones some of the bits in exactly one pile will change.

The winner of the game will always be the person who changes all piles to be 0. In the step that all piles become 0, the xor of the game also becomes 0. So if Alice can guarantee that she, and only she, always changes the xor of the game to be 0 then she can guarantee that she will win the game of Nim.

If the initial xor of the game is non-zero then the first player can always change the xor of the game to be zero, and thus win the game. If the xor of the initial state is already zero, then the second player can always change the xor of the game to zero and win. This is what motivates the algorithm for determining the winner of a particular game of Nim.

### *Grundy Numbers*





# *Index*

game theory, [11](#)  
graphs, [9](#)  
Grundy numbers, [12](#)

lists, [5](#)  
  array list, [5](#)  
  linked list, [5](#)

nim, [11](#)  
trees, [7](#)