NIM Games

William Gasarch- U of MD

NIM Games

NIM(1,2,3):

NIM Games

NIM(1,2,3):

1. Player I and Player II are looking at n stones.

NIM Games

NIM(1,2,3):

1. Player I and Player II are looking at n stones.
2. They alternate removing 1,2, or 3 stones, Player I goes first.

NIM Games

NIM(1,2,3):

1. Player I and Player II are looking at n stones.
2. They alternate removing 1,2, or 3 stones, Player I goes first.
3. The first one who can't move loses.

NIM Games

NIM(1,2,3):

1. Player I and Player II are looking at n stones.
2. They alternate removing 1,2, or 3 stones, Player I goes first.
3. The first one who can't move loses.

We assume they both play perfectly. For which n does Player I win? For which n does Player II Win?

NIM Games

NIM(1,2,3):

1. Player I and Player II are looking at n stones.
2. They alternate removing 1,2, or 3 stones, Player I goes first.
3. The first one who can't move loses.

We assume they both play perfectly. For which n does Player I win? For which n does Player II Win?
Play with 10 stones on white board.

NIM Games-Win Table for 1,2,3

Win Table Key is that

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W													

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II												

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I											

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I										

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I									

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II								

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I							

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I						

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I					

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II				

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II	I			

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II	I	I		

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II	I	I	I	

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II	I	I	I	II

NIM Games-Win Table for 1,2,3

Win Table Key is that

- $W(0) = II$. If there are 0 stones then Player II wins.
- If from n Player I can get to a II-spot then Player I wins. Otherwise Player II wins.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	I	I	II	I	I	I	II	I	I	I	II

Player II wins iff $n \equiv 0 \pmod 4$.

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W															

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II														

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I													

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II												

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I											

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I	I										

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I	I	I									

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I	I	I	I								

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*							

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I	I	I	I	II	I						

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*					

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	II	I	II	I	I	I	I	II	I	II	I				

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*			

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*		

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	*I*	

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*

NIM Games-Win Table for 1,3,4

Work on the win table for 1,3,4 together. I give you 5 minutes

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	*I*	*II*

Player II wins iff $n \equiv 0, 2 \pmod 7$.

Spotting Win Patterns for $(1, n, n+1)$-NIM

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.
2) (1,3,4) NIM is a mod 7 pattern.

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.
2) (1,3,4) NIM is a mod 7 pattern.
3) (1,4,5) NIM is YOU FIND OUT

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.
2) (1,3,4) NIM is a mod 7 pattern.
3) (1,4,5) NIM is YOU FIND OUT
4) (1,5,6) NIM is YOU FIND OUT

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.
2) (1,3,4) NIM is a mod 7 pattern.
3) (1,4,5) NIM is YOU FIND OUT
4) (1,5,6) NIM is YOU FIND OUT
5) (1,6,7) NIM is YOU FIND OUT

Spotting Win Patterns for $(1, n, n+1)$-NIM

1) (1,2,3) NIM is a mod 4 pattern.
2) (1,3,4) NIM is a mod 7 pattern.
3) (1,4,5) NIM is YOU FIND OUT
4) (1,5,6) NIM is YOU FIND OUT
5) (1,6,7) NIM is YOU FIND OUT
n) (1,n,n+1) NIM is YOU FIND OUT

A Computer Can Generate The Win Table

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.
$W(0) = II$ (Player II wins if there are 0 stones.)

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.
$W(0) = II$ (Player II wins if there are 0 stones.)
For $n = 1$ to 1000

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.
$W(0) = II$ (Player II wins if there are 0 stones.)
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.
$W(0) = II$ (Player II wins if there are 0 stones.)
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

Do you see why this does not quite work?

A Computer Can Generate The Win Table

Lets say you want to find the Win table for $(1, 6, 7)$ up to 1000.

Lets write a program for it!

The following almost works.
$W(0) = II$ (Player II wins if there are 0 stones.)
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

Do you see why this does not quite work?

$n - 6$ or $n - 7$ might be < 0.
How to fix this?

The Mathematical Fix

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.

$W(0) = II$.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.

$W(0) = II$.

For $n = 1$ to 1000

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.

$W(0) = II$.

For $n = 1$ to 1000

$W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR $W(n-7) = II$.

$W(n) = II$ otherwise.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.
I suspect you don't know those languages.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.
I suspect you don't know those languages.
You suspect that I don't.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.
I suspect you don't know those languages.
You suspect that I don't. Actually Pascal was my first langauge.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.
I suspect you don't know those languages.
You suspect that I don't. Actually Pascal was my first langauge.
I would recommend still using Python but checking for if the index is negative.

The Mathematical Fix

For $i = -7 to 0$ $W(i) = I$.
$W(0) = II$.
For $n = 1$ to 1000
    $W(n) = I$ if $W(n-1) = II$ OR $W(n-6) = II$ OR
$W(n-7) = II$.
    $W(n) = II$ otherwise.

This works.

However,

Java does not allow negative indices in arrays.

In Python $W[-1]$ means wrap-around.

Google AI tells me that Ada, FORTRAN, Pascal, and APL allow negative indices in arrays.
I suspect you don't know those languages.
You suspect that I don't. Actually Pascal was my first langauge.
I would recommend still using Python but checking for if the index is negative.
You will be doing this on the next HW.

A Computer Can Play Perfectly NIM Perfectly

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W													

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II												

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I											

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II										

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I									

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I								

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I							

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*						

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*					

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*				

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*	*I*			

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*		

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	*I*

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha!

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	*II*	*I*	*II*	*I*	*I*	*I*	*II*	*I*	*II*	*I*	*I*	*I*	*I*

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

Player I is now looking at 8. By spots $W(8-1) = W(7) = II$ so removes 1.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

Player I is now looking at 8. By spots $W(8-1) = W(7) = II$ so removes 1.

etc.

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

Player I is now looking at 8. By spots $W(8-1) = W(7) = II$ so removes 1.

etc.

If Player I is in a position where he can't win he can still see if

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

Player I is now looking at 8. By spots $W(8-1) = W(7) = II$ so removes 1.

etc.

If Player I is in a position where he can't win he can still see if

A Computer Can Play Perfectly NIM Perfectly

We assume computer is Player I. Similar holds for if Comp was Player II.

We already saw that a computer program can generate the win table.

Using the win table the computer can play perfectly

Example using Win Table for $1, 3, 4$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
W	II	I	II	I	I	I	II	I	II	I	I	I	I

If $n = 12$ then Player I looks to see if any of $W(12-1) = II$ OR $W(12-3) = II$ OR $W(12-4) = II$.

Ah Ha! $W(9) = II$. So Player I removes 3.

Player II is looking at 9 and knows he can't win :-(. Lets say he removes 1.

Player I is now looking at 8. By spots $W(8-1) = W(7) = II$ so removes 1.

etc.

If Player I is in a position where he can't win he can still see if

NIM and AI

NIM and AI

1) It is easy to write a program that plays NIM perfectly

NIM and AI

1) It is easy to write a program that plays NIM perfectly
2) Want to TRAIN an AI to play NIM perfectly.

NIM and AI

1) It is easy to write a program that plays NIM perfectly
2) Want to TRAIN an AI to play NIM perfectly.
3) If you train it against a perfect player it might not do well since there are moves it will never see.

NIM and AI

1) It is easy to write a program that plays NIM perfectly
2) Want to TRAIN an AI to play NIM perfectly.
3) If you train it against a perfect player it might not do well since there are moves it will never see.
4) Train it against imperfect player. An x-player makes the right move x-percent of the time and otherwise makes a random move.

NIM and AI

1) It is easy to write a program that plays NIM perfectly
2) Want to TRAIN an AI to play NIM perfectly.
3) If you train it against a perfect player it might not do well since there are moves it will never see.
4) Train it against imperfect player. An x-player makes the right move x-percent of the time and otherwise makes a random move.
5) Research question: If an AI-player is going to face y-players then what is x so that the AI should train against x-players?