# High School Projects In Computer Science

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

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

1. Write a program that will, given $1, x, y$, generate WIN table for NIM$(1, x, y)$. (say up to 200). TEST on 1,2,3 and 1,3,4 and others.

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

1. Write a program that will, given $1, x, y$, generate WIN table for NIM$(1, x, y)$. (say up to 200). TEST on 1,2,3 and 1,3,4 and others.
2. Write programs that will, given a WIN table, find the mod and win condition. TEST on 1,2,3 and 1,3,4 and others.

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

1. Write a program that will, given $1, x, y$, generate WIN table for NIM$(1, x, y)$. (say up to 200). TEST on 1,2,3 and 1,3,4 and others.
2. Write programs that will, given a WIN table, find the mod and win condition. TEST on 1,2,3 and 1,3,4 and others.
3. Run the program on 1,3,4 and 1,4,5 and 1,5,6 and ETC. By looking at data find a statement like
   Player II wins $1, x, x+1$ IFF $n \equiv BLAH \pmod{BLAHBLAH}$.
   Might involve cases like $x$ even or odd.

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

1. Write a program that will, given $1, x, y$, generate WIN table for NIM$(1, x, y)$. (say up to 200). TEST on 1,2,3 and 1,3,4 and others.
2. Write programs that will, given a WIN table, find the mod and win condition. TEST on 1,2,3 and 1,3,4 and others.
3. Run the program on 1,3,4 and 1,4,5 and 1,5,6 and ETC. By looking at data find a statement like
   Player II wins $1, x, x+1$ IFF $n \equiv BLAH \pmod{BLAHBLAH}$.
   Might involve cases like $x$ even or odd.
4. Same for $1, x, x+2$ and others.

# NIM Project I: Spotting Win Patterns for 1-Pile NIM

1. Write a program that will, given $1, x, y$, generate WIN table for NIM$(1, x, y)$. (say up to 200). TEST on 1,2,3 and 1,3,4 and others.
2. Write programs that will, given a WIN table, find the mod and win condition. TEST on 1,2,3 and 1,3,4 and others.
3. Run the program on 1,3,4 and 1,4,5 and 1,5,6 and ETC. By looking at data find a statement like
   Player II wins $1, x, x+1$ IFF $n \equiv BLAH \pmod{BLAHBLAH}$.
   Might involve cases like $x$ even or odd.
4. Same for $1, x, x+2$ and others.
5. (Optional) Automate the process

# NIM Project II: Player Perfectly and NOT so Perfectly

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.
2. Have a parameter $p$ so that each turn it makes the best move with prob $p$.

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.
2. Have a parameter $p$ so that each turn it makes the best move with prob $p$.
3. **Example** $n = 20$. Player II should win BUT Player II is an 0.8-player and Player I is an 0.9-Player. Have them play 100 times and see what the fraction of Player II wins are.

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.
2. Have a parameter $p$ so that each turn it makes the best move with prob $p$.
3. **Example** $n = 20$. Player II should win BUT Player II is an 0.8-player and Player I is an 0.9-Player. Have them play 100 times and see what the fraction of Player II wins are.
4. Conjecture: For $n = 20$ Player II will win ... some nontrivial amount of time. But as $n$ goes up and is still $\equiv 0 \pmod{4}$, Player II will win less and less.

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.
2. Have a parameter $p$ so that each turn it makes the best move with prob $p$.
3. **Example** $n = 20$. Player II should win BUT Player II is an 0.8-player and Player I is an 0.9-Player. Have them play 100 times and see what the fraction of Player II wins are.
4. Conjecture: For $n = 20$ Player II will win ... some nontrivial amount of time. But as $n$ goes up and is still $\equiv 0 \pmod 4$, Player II will win less and less.
5. More generally: $1, x, y$-NIM, Player I is a $p_1$-player, Player 2 is a $p_2$-player, $n$ stones, who wins? Most interesting case is when one player has positional adv and the other prob adv.

# NIM Project II: Player Perfectly and NOT so Perfectly

1. Write a program that plays (say) 1,2,3 NIM perfectly.
2. Have a parameter $p$ so that each turn it makes the best move with prob $p$.
3. **Example** $n = 20$. Player II should win BUT Player II is an 0.8-player and Player I is an 0.9-Player. Have them play 100 times and see what the fraction of Player II wins are.
4. Conjecture: For $n = 20$ Player II will win ... some nontrivial amount of time. But as $n$ goes up and is still $\equiv 0 \pmod 4$, Player II will win less and less.
5. More generally: $1, x, y$-NIM, Player I is a $p_1$-player, Player 2 is a $p_2$-player, $n$ stones, who wins? Most interesting case is when one player has positional adv and the other prob adv.
6. Very good for HS projects where the specs wants lots of stats like a science experiment.

# NIM Project III: AI and ML

**Note** I do not know any AI or ML but there are python packages that I've heard are good. Even so, if you do this YOU need to know or learn AI or ML or whatever.

# NIM Project III: AI and ML

**Note** I do not know any AI or ML but there are python packages that I've heard are good. Even so, if you do this YOU need to know or learn AI or ML or whatever.

1. A program is going to enter a NIM contest. What to train it against?

# NIM Project III: AI and ML

**Note** I do not know any AI or ML but there are python packages that I've heard are good. Even so, if you do this YOU need to know or learn AI or ML or whatever.

1. A program is going to enter a NIM contest. What to train it against?
2. If the program will play (say) Prob 0.8 Players then might not want to train it with perfect players since there are many positions they will never see.

# NIM Project III: AI and ML

**Note** I do not know any AI or ML but there are python packages that I've heard are good. Even so, if you do this YOU need to know or learn AI or ML or whatever.

1. A program is going to enter a NIM contest. What to train it against?
2. If the program will play (say) Prob 0.8 Players then might not want to train it with perfect players since there are many positions they will never see.
3. **Project** Answer the following question: If your AI is going to play against $p$-players then what is the $q$ such that you should train it against $q$-players.

# NIM Project IV: More Complicated NIM Games

Redo Projects 1,2,3 with more complicated NIM games.

# NIM Project IV: More Complicated NIM Games

Redo Projects 1,2,3 with more complicated NIM games.

1. 2-pile NIM. **Example:** Can remove 1,2,3 from pile 1 OR 1,3,4 from pile 2.

# NIM Project IV: More Complicated NIM Games

Redo Projects 1,2,3 with more complicated NIM games.

1. 2-pile NIM. **Example:** Can remove 1,2,3 from pile 1 OR 1,3,4 from pile 2.
2. many-pile NIM. You can imagine.

# NIM Project IV: More Complicated NIM Games

Redo Projects 1,2,3 with more complicated NIM games.

1. 2-pile NIM. **Example:** Can remove 1,2,3 from pile 1 OR 1,3,4 from pile 2.
2. many-pile NIM. You can imagine.
3. Nim With Cash-Bank Version: **Example** NIM(1,2,3). Player 1 begins with $x$ dollars, Player 2 with $y$ dollars. Each player can remove 1 or 2 or 3 stones. If a player removes $x$ stones he loses $x$ dollars to the bank.
4. Nim With Cash-Opponent Version: **Example** NIM(1,2,3). Player 1 begins with $x$ dollars, Player 2 with $y$ dollars. Each player can remove 1 or 2 or 3 stones. If a player removes $x$ stones he gives $x$ dollars to his opponent.

# Duels and Bullets

Alice and Bob both have guns.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.
3. Prob that Bob kills Alice is $p_B$.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.
3. Prob that Bob kills Alice is $p_B$.
4. Alice has $a$ bullets.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.
3. Prob that Bob kills Alice is $p_B$.
4. Alice has $a$ bullets.
5. Bob has $b$ bullets.

Given $(p_A, p_B, a, b)$ who has the advantage.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.
3. Prob that Bob kills Alice is $p_B$.
4. Alice has $a$ bullets.
5. Bob has $b$ bullets.

Given $(p_A, p_B, a, b)$ who has the advantage.
Involves some math, some recurrences.

# Duels and Bullets

Alice and Bob both have guns.

1. Alice and Bob alternate shooting each other
2. Prob that Alice kills Bob is $p_A$.
3. Prob that Bob kills Alice is $p_B$.
4. Alice has $a$ bullets.
5. Bob has $b$ bullets.

Given $(p_A, p_B, a, b)$ who has the advantage.
Involves some math, some recurrences.
Code up, collect data, make conjectures.

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$
Assume that $A$ are the only numbers you know about.

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$
Assume that $A$ are the only numbers you know about.
What numbers are prime?

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$
Assume that $A$ are the only numbers you know about.
What numbers are prime?
**Example** 9 is prime. Note that 3 is NOT in $A$.

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$
Assume that $A$ are the only numbers you know about.
What numbers are prime?
**Example** 9 is prime. Note that 3 is NOT in $A$.
**Project**

# Primes in Other Domains

Let
$A = \{1, 5, 9, 13, \ldots\}$
Assume that $A$ are the only numbers you know about.
What numbers are prime?
**Example** 9 is prime. Note that 3 is NOT in $A$.
**Project**
Look at this and other domains to investigate
How many primes are $\leq n$? How does that compare to the normal numbers?
Do we have Unique Factorization?

# Recurrences

Consider the following two recurrences

$a_1 = 1$

$a_n = a_{\lfloor \sqrt{n} \rfloor} + \lfloor \lg n \rfloor$

# Recurrences

Consider the following two recurrences

$a_1 = 1$

$a_n = a_{\lfloor \sqrt{n} \rfloor} + \lfloor \lg n \rfloor$

and

$a_1 = 1$

$a_n = a_{\lfloor \lg n \rfloor} + \lfloor \sqrt{n} \rfloor$

# Recurrences

Consider the following two recurrences

$a_1 = 1$

$a_n = a_{\lfloor \sqrt{n} \rfloor} + \lfloor \lg n \rfloor$

and

$a_1 = 1$

$a_n = a_{\lfloor \lg n \rfloor} + \lfloor \sqrt{n} \rfloor$

which one grows faster? What do they look like? What matters more the subscript or the additive term?

# Recurrences

Consider the following two recurrences

$a_1 = 1$

$a_n = a_{n-1} + a_{\lfloor n/2 \rfloor}$

# Recurrences

Consider the following two recurrences

$a_1 = 1$

$a_n = a_{n-1} + a_{\lfloor n/2 \rfloor}$

**Known** If look at the equation mod 2,3,5,7 then it hits 0 infinitely often.

**Open** What about other mods?

**Project** Gather evidence for conjectures. Also vary initial conditions.

# SAT Solvers

Given a boolean formula like

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4)$$

We want to know if it is SATISFIABLE. There are many algorithms for this. Code them up, see how they do.

# Crypto

William Gasarch- U of MD

# Crypto Projects

(1) Coding up various encryption, decryption, and cracking algorithms.

# Crypto Projects

(1) Coding up various encryption, decryption, and cracking algorithms.
(2) Coding up secret sharing and its variants.

# Crypto Projects

(1) Coding up various encryption, decryption, and cracking algorithms.
(2) Coding up secret sharing and its variants.
(3) Coding up factoring algorithms and RSA. (Quad Sieve possibly.)

# Crypto Projects

(1) Coding up various encryption, decryption, and cracking algorithms.
(2) Coding up secret sharing and its variants.
(3) Coding up factoring algorithms and RSA. (Quad Sieve possibly.)
(4) Coding up discrete log and Diffie-Helman.

# Complexity Theory

(Needs background so you might not understand this slide.)

# Complexity Theory

(Needs background so you might not understand this slide.)
(1) Code up SAT Solvers. Apply VV to it.

# Complexity Theory

(Needs background so you might not understand this slide.)
(1) Code up SAT Solvers. Apply VV to it.
(2) DFA-tricks for division VS just doing the division.

# Complexity Theory

(Needs background so you might not understand this slide.)
(1) Code up SAT Solvers. Apply VV to it.
(2) DFA-tricks for division VS just doing the division.
(3) Do more queries help: 3-SUM, APSP.

# Coloring and Functions

Let $f$ be a function from $N$ to $N$ (we take $N$ to NOT have 0). We use $f$ to color the integers.

- Color $n$ **RED** if it is the value of $f$ on a **BLUE** number
- Color $n$ **BLUE** otherwise.

# Coloring and Functions

Let $f$ be a function from $N$ to $N$ (we take $N$ to NOT have 0). We use $f$ to color the integers.

- Color $n$ **RED** if it is the value of $f$ on a **BLUE** number
- Color $n$ **BLUE** otherwise.

**Example** $f(x) = x + 1$.
1 is NOT in the $f$ of ANYTHING so its NOT $f$ on a **BLUE** .
Hence $COL(1) = BLUE$.
2 IS $f(1)$ and 1 is **BLUE** , so 1 is **RED**
3 is $f(2)$ but 2 is **RED** , so 3 is **BLUE**
$\cdots$.
So we get an alternating pattern.
What happens with other functions? With more complicated rules?