

**Universidade Federal do Ceará**  
**Departamento de Computação**  
**Disciplina: Inteligência Artificial**  
**Prof. João Paulo do Vale Madeiro**

**Aula Prática 02 – Busca para Jogos**













1 – *Nim* é um jogo para dois jogadores, que sempre termina com a vitória de um jogador. O jogo consiste em várias fichas colocadas sobre a mesa de jogo e os jogadores se revezando na remoção de uma ou mais fichas.

Considere as seguintes regras:







- Existem vários “contadores” em uma pilha compartilhada;
- Dois jogadores realizam movimentos alternados;
- Na sua vez, um jogador remove uma, duas ou três peças da pilha;
- O jogador que fizer o último lance perde o jogo.

Para demonstrar as regras, dois jogadores – Mindy e Maximillian – jogarão uma partida de Simple-Nim, começando com doze “contadores”. Mindy vai primeiro:

- Mindy pega duas peças, deixando dez na pilha.
- Maximillian pega uma peça, deixando nove na pilha.
- Mindy pega três peças, deixando seis na pilha.
- Maximillian pega duas peças, deixando quatro na pilha.
- Mindy pega três peças, deixando uma na pilha.
- Maximillian pega o último contra-ataque e perde o jogo.

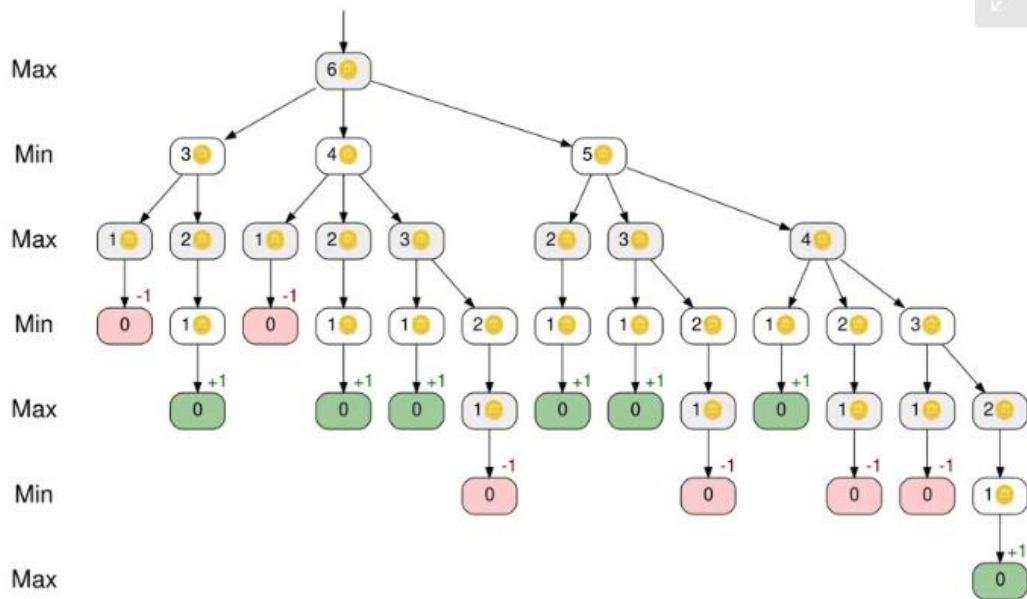
Mindy	Pile	Maximillian
		
		
		
		
		
		
		

Esta representação do jogo mostra explicitamente quantas peças cada jogador removeu em seu turno. No entanto, há algumas informações redundantes na tabela. Você pode representar o mesmo jogo apenas controlando o número de peças na pilha e de quem é a vez:

Player to move	Pile
Mindy	 (12)
Maximillian	 (10)
Mindy	 (9)
Maximillian	 (6)
Mindy	 (4)
Maximillian	 (1)

Podemos representar o estados do jogo também do seguinte modo: 12-10-9-6-4-1

Podemos representar uma árvore de busca para o problema e implementar o algoritmo MinMax, a partir do estado em que restam 6 peças no jogo:



Implementar um algoritmo Minimax em Python específico do Nim

Considere o mesmo exemplo da seção anterior: é a vez de Maximiliano e há seis fichas na mesa. Você usará o algoritmo minimax para confirmar se Maximillan pode vencer este jogo e calcular seu próximo movimento.
















## 2) Jogue o jogo normal de Nim

É hora de retirar as regras regulares do Nim. Você ainda pode reconhecer o jogo, mas permite aos jogadores um pouco mais de escolha:

- Existem várias pilhas, com vários contadores em cada uma.
- Dois jogadores realizam turnos alternados.
- Na sua vez, o jogador remove quantas fichas quiser, mas as fichas devem vir da mesma pilha.
- O jogador que fizer o último contador perde o jogo.

Observe que não há mais nenhuma restrição sobre quantos marcadores remover em um turno. Se uma pilha contiver vinte fichas, o jogador atual poderá pegar todas elas.

Por exemplo, considere um jogo que começa com três pilhas contendo duas, três e cinco fichas, respectivamente. Veja como os jogadores Mindy e Maximillian jogam:

Player to move	Pile one	Pile two	Pile three
Mindy			
Maximillian			
Mindy			
Maximillian			
Mindy			
Maximillian			

Mindy pega quatro fichas da terceira pilha, deixando duas, três e uma fichas.

Maximillian pega duas fichas da segunda pilha, deixando duas, uma e uma ficha.

Mindy pega uma ficha da primeira pilha, deixando uma, uma e uma ficha.

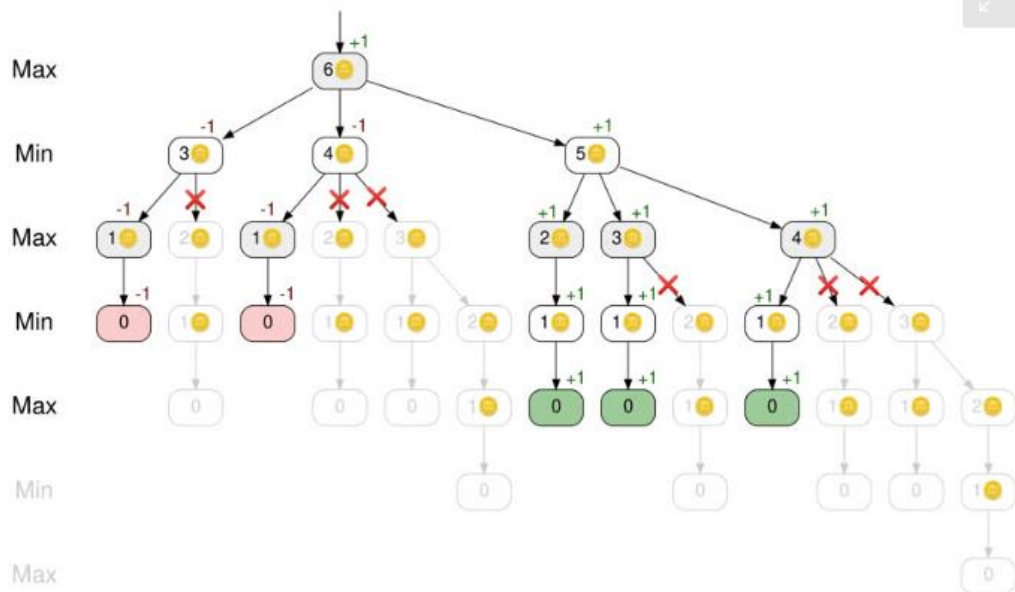
Maximillian não tem mais nenhuma escolha interessante, mas pega uma ficha da terceira pilha, deixando uma, uma e zero fichas.

Mindy pega uma ficha da segunda pilha, deixando fichas um, zero e zero.

Maximillian pega o último contador restante e perde o jogo.

Implemente um algoritmo MinMax para o desenvolvimento da melhor jogada de acordo com uma dada configuração apresentada.

**3) Realize a implementação da poda, conforme esquema exemplificativo abaixo:**



4) Usar o algoritmo MinMax para calcular o melhor movimento de uma peça simples no xadrez (como um Peão ou uma Rainha) em um tabuleiro. O exercício envolve entender o funcionamento básico do MinMax, aplicando-o a um jogo mais complexo.

Implemente o movimento de uma peça (por exemplo, um peão ou uma rainha) em um tabuleiro de xadrez 8x8. O tabuleiro pode ser simplificado para se concentrar apenas em uma peça jogando contra outra.

A IA deve usar o MinMax para escolher o melhor movimento para a peça de acordo com a situação no tabuleiro, levando em consideração as jogadas possíveis do oponente.

Use a função de avaliação de tabuleiro, considerando se uma peça do oponente pode ser capturada no próximo movimento ou se uma jogada resulta em uma ameaça direta.

Implemente um pequeno número de movimentos (ex.: até 3 movimentos para simplificar a análise MinMax).