PROJETO ALGORITMOS E ESTRUTURAS DE DADOS (IF-969)

Por: Antônio (apln2)

JOGO DA VELHA

O Jogo da velha é um jogo e passatempo popular. É um jogo de regras extremamente simples, que não traz grandes dificuldades para seus jogadores e é facilmente aprendido. A origem é desconhecida, com indicações de que pode ter começado no antigo Egito, onde foram encontrados *tabuleiros* esculpidos na rocha, que teriam mais de 3.500 anos.

Por sua simplicidade, o jogo da velha é um jogo fácil de ser dominado e o resultado mais comum obtido em partidas é o empate. Para eliminar esse problema, surgiram ao longo do tempo diversas versões do jogo da velha, como o jogo miserè, o jogo selvagem e até o jogo da velha quântico, onde os movimentos dos jogadores são superposições das peças ‘X’ e ‘O’.

ÁRVORE DE JOGO

Uma árvore de jogo é um grafo árvore direcionado onde cada vértice é uma posição possível e cada aresta é um movimento. O vértice de índice 0 do grafo é equivalente a disposição inicial das peças de jogo e seus adjacentes são as posições possíveis para as peças depois de um movimento.

Árvores de jogos são importantes para a inteligência artificial, pois uma forma de se determinar o próximo movimento em um jogo é a partir da execução de algum algoritmo de busca a partir de uma subárvore local.

Em uma árvore de jogo, o número de folhas é equivalente ao número de formas possíveis que o jogo pode acabar. Apenas para se ter uma ideia, em um jogo simples como o jogo da velha, o número de folhas é igual a 255.168. No xadrez, por exemplo, estima-se que o número de jogadas possíveis em um jogo de Xadrez seja de 10^{120}. Fica claro, então que a elaboração de uma árvore de jogo completa é um processo custoso.

Partindo deste ponto, apresento o projeto da unidade.

O PROJETO

Nesta unidade, o projeto será a construção de um Algoritmo que joga versões do jogo da velha. Seu algoritmo deve usar um grafo-árvore direcionado e ponderado para a simulação de sua árvore de jogo.

Considere que:

- O vértice de índice 0 deve ser equivalente à posição inicial do tabuleiro;

- Seu algoritmo deve implementar os algoritmos de busca em largura, busca em profundidade e menor caminho;
- Sempre que o algoritmo representar um movimento que não conhecia antes, um novo vértice deve ser criado na árvore de jogo e, a partir daí, as jogadas devem ser aleatórias;
- Para calcular seu próximo movimento, o algoritmo deve calcular o menor caminho possível até um vértice que resulte em vitória, ou até na direção do caminho mais curto até o enésimo vértice de sua posição, sendo n o número de arestas que separa o vértice atual do enésimo vértice. Este n deve ser um dos parâmetros para a inicialização da sua classe GameTree.
- Todas as arestas do grafo possuem peso inicial 1. Este peso é incrementado em 1 para a quantidade de vértices que resultam em derrota na subárvore seguinte a essas arestas e decrementado em 1 para a quantidade de vitórias, sendo 1 o peso mínimo da aresta;
- O cálculo do peso da aresta é feito dinamicamente por meio de um algoritmo de busca em profundidade;
- Os vértices devem representar APENAS posições válidas.

O JOGO

Seu algoritmo deve implementar a construção de árvores de jogo de duas variações do jogo da velha, sendo a primeira, e obrigatória, a versão **Notakto** e a segunda a ser escolhida entre as versões abaixo:

1. **Jogo Selvagem (wild tic-tac-toe)**;
2. **Miserè**;
3. **Super tic-tac-toe**.

ESTRATÉGIA DE TREINAMENTO

Seus algoritmos devem ser capazes de se comunicar uns com os outros e consigo mesmo, devendo implementar uma estratégia de comunicação onde, ao fazer uma jogada o algoritmo tenha como output um código reconhecível para a jogada e, receba como input ou parâmetro a jogada do adversário.

PERSISTÊNCIA

Por fim, seu algoritmo deve ser capaz de salvar sua árvore de jogo atual em um arquivo ‘game-tree.txt’ e, a partir desse arquivo, carregar novamente sua árvore de jogo, quando reinicializado.

REGRAS

1. O trabalho pode ser feito individualmente ou em dupla;
2. O prazo de entrega final é até às 23:59 do dia 19/06/2019;
3. Você deve entregar todos os arquivos usados na construção do seu algoritmo anexando-os todos na mesma entrega. Entrega de arquivos comprimidos (especialmente arquivos .rar) resultarão na dedução de pontos.
4. Entre os arquivos entregues, deve haver pelo menos um arquivo 'game-tree.txt' de cada um dos seus jogos implementados;
5. É necessária apenas uma entrega por dupla. Os arquivos devem conter cabeçalho contendo o nome dos responsáveis, e-mail e login (veja exemplos de cabeçalhos no repositório da monitoria);
6. Com exceção das biblioteca numpy, sys e time, NENHUMA outra biblioteca pode ser utilizada.
7. Plágio total ou parcial e falta de ética implica na desistência dos pontos.

REFERÊNCIAS

JOGO da velha. In. Wikipédia: a enciclopédia livre. Disponível em: <https://pt.wikipedia.org/wiki/Jogo_da_velha>. Acesso em: 29 abr 2019.

NOTAKTO. In. Wikipédia: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Notakto>. Acesso em 29 abr 2019.

TIC-TAC-TOE (with Xs only). [*S. l.*]: Numberphile, 2015. Disponível em: <https://www.youtube.com/watch?v=ktPvjr1tiKk>. Acesso em: 29 maio 2019.

TIC-TAC-TOE variants. In. Wikipédia: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Tic-tac-toe_variants>. Acesso em: 29 abr 2019.

GAME tree. In. Wikipédia: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Game_tree>. Acesso em: 29 abr 2019.