

Inteligência Artificial

Trabalho de Programação 1: Solução de Problemas por Busca

Pontuação: 30 pontos + 8 pontos extras

Data de Entrega: 13/06/2022

Informações básicas:

- O trabalho deve ser submetido até a data limite na atividade do AVA.
- Deve ser submetido um arquivo compactado contendo os programas desenvolvidos e o relatório em formato pdf.
- É permitido que os alunos conversem a discutir suas soluções, mas os códigos-fonte devem ser produzidos individualmente. Se forem identificados casos de cola, serão abertos processos que podem culminar no desligamento do programa de mestrado.

Descrição do Trabalho:

O trabalho consiste em implementar e comparar a performance dos algoritmos de busca *depth-first search*, *uniform cost search* e *A**. Para ajudá-los a começar e prover um ambiente para testes dos algoritmos, é dada uma implementação do algoritmo *breadth-first search* para busca de caminho em um labirinto em anexo à esta especificação.

Para o desenvolvimento do trabalho não é permitido o uso de bibliotecas que implementem os algoritmos, mas é permitido usar bibliotecas auxiliares, e.g., que implementem estruturas de dados. Os algoritmos de busca podem ser úteis em entrevistas de programação em empresas de grande porte, então considere este tempo de implementação como um investimento na sua carreira.

Para comparar os algoritmos, deve ser usada um labirinto com tamanho 300x300 e as métricas de comparação são tempo de execução, número de nós expandidos, número de nós gerados, custo do caminho e tamanho do caminho. Ao medir o tempo de execução, desligue a visualização porque o custo de atualização da visualização é maior do que o custo do algoritmo. Tome o cuidado de usar o mesmo labirinto em todos os casos (e.g., fixe o parâmetro seed da classe MazeProblem).

Em adição aos códigos, deve ser escrito um relatório curto (1-3 páginas) com a estrutura abaixo:

- Fundamentação Teórica: Descrever os algoritmos que serão implementados e suas diferenças.
- Experimentos: Descrever o experimento que será realizado e os resultados esperados considerando a teoria. Incluir a configuração do computador que será usado nos experimentos.
- Resultados: Apresentar a comparação dos algoritmos como uma tabela em que linhas são os algoritmos e colunas são as métricas. Discutir se os resultados foram consistentes com o esperado pela teoria. Se não, apresentar hipóteses do porquê.

Pontuação:

- Implementação do depth-first search: 5 pontos.
- Implementação do uniform cost search: 5 pontos.
- Implementação do A*: 15 pontos.
- Relatório: 5 pontos.

Pontos Extras:

- Implementar e usar um Heap de Fibonacci para a fila de prioridade no A* e uniform cost search: +3 pontos.
 - A implementação pode ser baseada em códigos disponíveis na internet.
 - **Atenção:** Se for encontrado um caminho de menor custo para um nó, a sua posição na fila deverá ser atualizada. Não serão consideradas soluções que mantenham duas cópias do nó no heap com custos diferentes.
 - O código deve ser comentado para explicar o funcionamento do heap.
- Implementar um problema adicional (e.g., n rainhas) e adicionar a comparação dos algoritmos neste problema ao relatório: +3 pontos.
 - Deve ser criada uma visualização, mesmo que simplificada, que permita acompanhar a evolução do algoritmo.
 - Os problemas devem ser minimamente sofisticados. O ambiente do robô de limpeza com duas células não é válido. Em caso de dúvida, converse com o professor.
- Implementar o *iterative deepening depth-first search*: +2 pontos.