

 <p>INSTITUTO FEDERAL ESPIRITO SANTO Campus Serra</p>	<p style="text-align: center;">IFES</p> <p style="text-align: center;">Inteligência Artificial</p> <p style="text-align: center;">Exercício de Programação 1: Solução de Problemas por Busca</p>	<p style="text-align: center;">Nota</p>
--	--	---

Professor: Sérgio Nery Simões

Data: 05/05/2023

Nome: _____

Turma: _____

Informações:

- O trabalho deve ser submetido até a data limite na atividade do AVA.
- Deve ser submetido um arquivo compactado contendo o código em Python dos programas desenvolvidos e o relatório em formato pdf.
- É permitido que os alunos conversem e discutam 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 aluno.

Descrição do Trabalho

Questão 1 (Q1)

Implementar e comparar o desempenho dos algoritmos de busca: **(i) Depth-first search**, **(ii) Uniform cost search** e **(iii) A* search**. 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 ajudá-los a começar e prover um ambiente para testes dos algoritmos, é fornecida uma implementação do algoritmo *breadth-first search* para busca de caminho em um labirinto em anexo à esta especificação. Você deve utilizar a estrutura de dados fornecida por essa implementação para implementar os demais algoritmos solicitados neste trabalho.

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 (ex: NetworkX, etc).

Para comparar os algoritmos, deve ser usado um labirinto com tamanho **300x300** com percentual de bloqueio 50%, e as métricas de comparação são:

1. tempo de execução;
2. número de nós expandidos;
3. número de nós gerados;
4. custo do caminho e
5. tamanho do caminho.

Você deverá criar uma Tabela comparativa com os algoritmos de busca (nas linhas) e as respectivas métricas resultantes (nas colunas) para cada algoritmo.

Ao medir o tempo de execução, desligue a visualização porque o custo de atualização da visualização é maior que o custo do algoritmo. Para a comparação ser justa, tome o cuidado de usar o mesmo labirinto em todos os casos (e.g., fixe o parâmetro seed=42 da classe MazeProblem).

Bom trabalho!

Questão 2 (Q2)

Implementar os algoritmos UCS e A* *search* para o problema de roteamento entre cidades descrito no livro-texto da disciplina. O roteamento deve ser feito entre as cidades 'Arad' e 'Bucharest'. Para auxiliá-los, é fornecida uma implementação inicial do algoritmo no link abaixo:

<https://colab.research.google.com/drive/15iUnVYFc5uA-Q2SQS7VpzQfAIO4r6ASX>

Compare os resultados obtidos pela aplicação dos algoritmos UCS e A* *search* em relação ao (i) caminho obtido e (ii) custo do caminho através de uma tabela.

Relatório

Em adição aos códigos, deve ser escrito um **relatório curto (3-4 páginas)** com a estrutura abaixo:

- **Fundamentação Teórica:** Descrever brevemente os algoritmos implementados e suas diferenças. Faça uma tabela com as complexidades dos algoritmos BFS, DFS, UCS e A*.
- **Experimentos:** Descrever como foi realizado o experimento 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:

Atividade	Pontuação
[Q1] Implementação do depth-first search:	15 pontos.
[Q1] Implementação do uniform cost search:	15 pontos.
[Q1] Implementação do A*:	30 pontos.
[Q2]: Adaptação do UCS e A* ao roteamento:	20 pontos.
Relatório sobre Q1 e Q2:	20 pontos.

Pontos Extras:

1. Implementar o *iterative deepening depth-first search*: +10 pontos.
2. Implementar um problema adicional (e.g., n rainhas) e adicionar à comparação dos algoritmos neste problema ao relatório: +10 pontos.

Obs:

- 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.

Bom trabalho!