

Avaliação Intermediária – AI_a

25/03/2026

Aluno(a): _____

Curso: _____ Nº de matrícula: _____

Turma: _____ Professor(a): _____

Prezado(a) Aluno(a),

Você terá 120 minutos a partir do início oficial da prova para concluir esta avaliação, administre bem o seu tempo. Leia atentamente as instruções a seguir e as questões da prova antes de começar a resolvê-la.

- Identifique-se com letra legível em todas as folhas de prova.
- Esta avaliação é composta de 4 questões e um total de 7 páginas. Verifique se a prova está completa e/ou se há problemas de impressão e comunique o aplicador antes de iniciar a prova. Comunicação posterior não será considerada.
- Para a resolução das questões, utilize apenas os campos demarcados e não destaque as folhas de prova.
- A resolução da prova poderá ser feita a lápis ou a caneta. Avaliações feitas a lápis, no entanto, não serão revisadas pelo professor.
- Em caso de dúvida sobre alguma questão desta avaliação, redija um texto na folha de prova explicitando-a para que o professor avalie a pertinência durante a correção.
- Portar qualquer material estranho à folha de questões – celular (ligado ou desligado), tablet, notebook, anotações e livro – constituirá violação ao Código de Ética e de Conduta e acarretará sanções nele previstas.
- Consulta a colegas e a qualquer material estranho à folha de questões (celular, tablet, notebook, anotações e livro) constituirão violações ao Código de Ética e de Conduta e acarretarão sanções nele previstas. Faça o seu trabalho de maneira ética!
- Você somente poderá sair da sala depois de entregar a prova. Caso necessite sair durante a realização da avaliação, peça autorização antecipadamente ao aplicador.

Boa Prova!

Para uso exclusivo do Professor

Orientações durante a avaliação:

Permitido utilizar folha para rascunho/anotações?

(X) SIM () NÃO - deve anexar ao material da avaliação

Permitido realizar consulta de material?

() SIM (X) NÃO

Avaliação passível de revisão?

(X) SIM () NÃO

Questões

*Quaisquer hipóteses relevantes devem ser **explicitamente formuladas**. Faz parte da avaliação da prova a **correta interpretação** das questões. A **clareza** e a **objetividade** das respostas serão consideradas na avaliação.* Sugestão: leia todas as questões primeiro antes de responder qualquer questão da avaliação.

Questão 1.

(1 ponto) Qual é a definição de agente autônomo? Qual é a relação deste conceito com Inteligência Artificial?

Questão 2.

Considere um agente que está inserido em um ambiente do tipo *GridWorld* com dimensão de 2 colunas e 3 linhas. Este agente sabe executar as ações: ir para cima, ir para baixo, ir para esquerda e ir para direita. O objetivo deste agente é passar por todas as células do *GridWorld*. Ou seja, trata-se de um problema de *Coverage Path Planning*, um problema onde o agente precisa passar por uma determinada área o mais rápido possível. Este tipo de comportamento é útil em diversos cenários, como por exemplo, robôs aspiradores de pó e drones para busca e resgate em alto mar. Neste *GridWorld* pode existir uma única célula que é um obstáculo, ou seja, o agente não consegue chegar nela. O agente inicia em uma célula qualquer. O objetivo dele é criar um plano para passar por todas as células o mais rápido possível. Ele não precisa voltar para a célula de onde ele saiu. O estado meta para o agente é quando ele alcança a última célula não vista.

(1 ponto) Defina como o estado deve ser representado. Deixe claro qual a estrutura de dados que deverá ser usada, assim como o seu significado. Lembre-se de representar somente aquilo que é necessário para resolver o problema. Depois de definida a forma de representação do estado, forneça um exemplo de estado inicial possível e um exemplo de estado final usando a representação adotada. Ao final, defina o método **init** da classe em **python**.

(1 ponto) Defina o método para geração de sucessores. Lembre-se que as ações são 4: ir para esquerda, ir para a direita, ir para cima e ir para baixo. Defina o método para geração de sucessores em **python** e que seja coerente com a estrutura de dados utilizada para representar o estado.

(1 ponto) Defina o método `is_goal(self)` para o problema.

(1 ponto) Qual é a ramificação do problema? Qual é a estimativa de profundidade da solução? Qual é a estimativa do tamanho da árvore de busca?

(1 ponto) Estamos interessados apenas em soluções ótimas. Por enquanto, não temos nenhuma preocupação com espaço e tempo. Sendo assim, liste os algoritmos que podemos utilizar.

(1 ponto) Defina uma heurística que é admissível e não do contra (que realmente ajuda a encontrar uma solução rapidamente) para este problema.

Questão 3.

(2 pontos) Considere o grafo da figura 1, onde os números nas arestas representam os custos de transição entre os nós. O nó **E** é o estado meta. Qualquer nó no grafo pode ser o estado inicial.

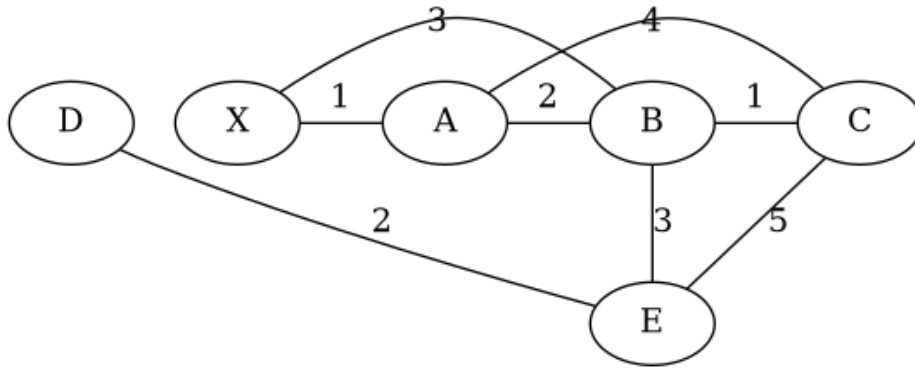


Figure 1: Grafo da questão 3

Você deve preencher a tabela abaixo com os valores de $h_1(N)$, $h_2(N)$ e $h_3(N)$ para cada nó N do grafo, onde $h_1(N)$ é a heurística admissível e não do contra, $h_2(N)$ é uma heurística admissível mas que não ajuda muito na busca, e $h_3(N)$ é uma heurística não admissível.

Nó	$h_1(n)$	$h_2(n)$	$h_3(n)$
A			
B			
C			
D			
E			
X			

Questão 4.

(1 ponto) Os algoritmos vistos em sala de aula foram: Busca em Largura (BL), Busca em Profundidade (BP), Busca em Profundidade Iterativa (BPI), Busca de Custo Uniforme (BCU), Busca Gananciosa (BG) e A^* . Todos eles tem uma forma de funcionamento muito parecida, com um código muito parecido. Os únicos que são ligeiramente diferentes são o BP porque tem um controle adicional sobre o limite m e o BPI porque reutiliza o BP. Os demais algoritmos, BL, BCU, BG e A^* tem uma implementação muito parecida:

```
function [____(1)____] (Estado inicial): Nodo  
  [____(2)____] abertos  
  abertos.add(new Nodo(inicial))  
  while abertos.size() > 0 do  
    Nodo  $n \leftarrow$  abertos.removeFirst()  
    if  $n.getEstado().éMeta()$  then  
      return  $n$   
    end if  
    abertos.append([____(3)____])  
  end while  
  return null
```

Para cada um dos algoritmos listados, preencha os espaços em branco (1,2,3) no pseudo-código apresentado acima.

(1)	(2)	(3)
BL		
BCU		
BG		
A^*		