

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Curso de Ciência da Computação - Coração Eucarístico

Profa.: Camila Laranjeira - mila.laranjeira@gmail.com

Disciplina: Inteligência Artificial / 1o Semestre de 2022

Aluna(o): Matheus Rangel de Figueiredo

Exercício Prático 01 - Pacman #1

Instruções:

- Consulte os slides da disciplina para maiores detalhes sobre a implementação
- O código fonte base está no Canvas sob o título pacman.zip
- Você deve entregar seu código em um .zip que inclua esse documento preenchido

1. Implemente a **busca em profundidade** e a **busca em largura** para solucionar os cenários do Pacman apresentados na tabela a seguir. Reporte seus resultados de acordo com as informações apresentadas no terminal ao final de cada execução (vide figura).

```
Path found with total cost of 10 in 0.0 seconds
Search nodes expanded: 17
Pacman emerges victorious! Score: 500
Average Score: 500.0
Scores:      500.0
Win Rate:    1/1 (1.00)
Record:      Win
```

Busca	Labirinto	Custo Total	É ótimo?	Tempo (s)	#Nós
DFS	tinyMaze	10	sim	0.0 seconds	15
BFS	tinyMaze	8	sim	0.0 seconds	15
DFS	mediumMaze	130	nao	0.0 seconds	146
BFS	mediumMaze	68	sim	0.0 seconds	269
DFS	bigMaze	210	nao	0.0 seconds	390
BFS	bigMaze	210	sim	0.0 seconds	620

1.1 O sistema colore de vermelho todos os nós expandidos durante a busca, com a cor mais forte quanto mais cedo o nó foi visitado (vide figura). Analise suas execuções do DFS e BFS para o `tinyMazeSearch` e responda:

- A ordem de visita da sua solução faz sentido para cada algoritmo? Por que?
- O Pacman caminha por todos os nós explorados na hora de jogar? Por que?



Sim, pois ele consegue identificar o custo, o tempo e os nós visitados.

Sim, pois ele procura o melhor caminho baseado no algoritmo implementado em cada função.

1.2 Reporte a complexidade de tempo e espaço para as suas implementações do DFS e BFS. As complexidades são iguais no pior caso, melhor caso e caso médio? Por que?

Pior caso (espaço) -> DFS = $O(D)$, BFS = $O(b^d)$
Melhor caso (espaço) -> DFS = $O(1)$, BFS = $O(1)$
Caso médio (espaço) ->
Pior caso (tempo) -> DFS = $O(D)$, BFS = $O(b^d)$
Melhor caso (tempo) -> DFS = $O(1)$, BFS = $O(1)$
Caso médio (tempo) ->

2. Implemente a busca de custo uniforme e a busca A* para solucionar o cenário do bigMaze e realize as seguintes comparações.

Busca	H(n)	Labirinto	Custo Total	É ótimo?	Tempo (s)	#Nós
UCS	-	bigMaze				
A*	Manhattan	bigMaze				
A*	Euclidiana	bigMaze				
A*	Customizada*	bigMaze				

*opcional

21 Use o espaço abaixo para descrever as suas impressões sobre os resultados obtidos. Inclua tudo que achar relevante, desde as diferenças na implementação até o impacto de cada variação nos experimentos observados. Caso tenha definido uma heurística customizada, descreva aqui.

3. Execute cenários onde a função de custo do caminho não é constante. Lembre-se, o UCS é uma busca cega, e portanto não avalia a qualidade de nós não-terminais, **se baseando apenas na função de custo.**

Busca	Player	Labirinto	Score	#Nós	Custo total
UCS	StayEastSearchAgent	mediumDottedMaze			
UCS	StayWestSearchAgent	mediumDottedMaze			
UCS	StayEastSearchAgent	mediumScaryMaze			
UCS	StayWestSearchAgent	mediumScaryMaze			

3.1 Use esse espaço para justificar o que influenciou as variações da tabela acima, destacando principalmente quais fatores impactam cada uma das colunas (score, #nós explorados, e custo total do caminho selecionado).