

## **Relatório EP - MAC0425**

**Lucas Quaresma Medina Lam**

**NUSP: 11796399**

**Roberto Oliveira Bolgheroni**

**NUSP: 11796430**

### **Questão 1:**

Sim, foi de acordo com o esperado. A busca em profundidade explora os estados utilizando uma pilha LIFO, dessa forma muitos nós iniciais são ignorados, considerando que os nós escolhidos encontram uma solução, ao final. Além disso, os últimos nós explorados são os finais, pois ao achar uma solução, o algoritmo termina.

Nem todos os estados explorados são utilizados no caminho escolhido. Principalmente no final, no labirinto “bigMaze”, por exemplo, alguns caminhos explorados são “sem-saída” e como o algoritmo utiliza backtrack, esses caminhos são descartados. Apenas o caminho que leva ao destino é considerado como a solução final.

### **Questão 2**

Em geral, para a busca DFS, a solução encontrada não é necessariamente ótima. No caso do “mediumMaze”, a solução encontrada de fato não é ótima. A busca em profundidade sempre termina quando encontra um caminho que leva até o objetivo, dado pelo aprofundamento. Sendo assim, não há garantia que o caminho encontrado seja o de menor comprimento.

A busca em profundidade não considera o custo/comprimento de cada caminho, e não faz comparação entre os caminhos já percorridos, retornando sempre o primeiro encontrado.

### **Questão 3**

A solução encontrada pela BFS é ótima. A busca em largura expande todos os nós de uma certa altura da árvore antes de passar para o próximo, e como todas as ações possuem o mesmo custo, o custo de um caminho se torna seu comprimento. Dessa forma, a BFS encontrará a solução de menor comprimento primeiro, a qual por consequência será a de menor custo.

### **Questão 4**

Para o labirinto “mediumMaze”, a solução pela busca UCS é igual a da BFS, mas diferente da DFS. Como todas as ações possuem o mesmo custo (1), e o custo de uma solução se torna o seu comprimento, a busca BFS encontra sempre as mesmas soluções da UCS, já que o custo de um caminho coincide com seu comprimento.

Já a DFS não faz distinção de comprimento/custo de uma solução; ela explora ramos da árvore de busca à exaustão até encontrar um caminho ao objetivo. Há casos em que coincidem as soluções dos três métodos de busca, como para o labirinto “bigMaze”, mas no caso geral, apenas UCS e BFS coincidem sempre.

## Questão 5

O algoritmo A\* encontra uma solução mais rapidamente que as outras buscas pois combina estratégias para escolher um nó que se espera que esteja mais próximo da solução, utilizando a função de custo somada de uma heurística consistente, assim expandindo primeiro os nós que têm maior chance de levar ao objetivo e evitando aqueles que demorariam mais, fazendo com que encontre a solução de maneira mais efetiva.

Se tivéssemos uma heurística inconsistente a busca A\* pode escolher nós que não estão tão próximos da solução, expandindo-os e assim se tornando ineficiente comparado a uma heurística consistente. No geral também, não se pode garantir que uma heurística inconsistente irá encontrar o caminho ótimo (como no caso de busca em grafo).

## Questão 6

Como esperado, tanto os nós explorados quanto os caminhos finais encontrados, que são ótimos, são iguais para as buscas BFS e UCS. Curiosamente, para o A\*, os resultados também são idênticos. A única diferença entre os três é no tempo de busca; o BFS foi mais rápido que os outros dois.

Já o DFS expande um número maior de nós e encontra uma solução pior que as dos outros métodos de busca.

DFS:

Path found with total cost of 298 in 0.0 seconds  
Search nodes expanded: 806  
Pacman emerges victorious! Score: 212

BFS:

Path found with total cost of 54 in 0.0 seconds  
Search nodes expanded: 682  
Pacman emerges victorious! Score: 456

UCS:

Path found with total cost of 54 in 0.1 seconds  
Search nodes expanded: 682  
Pacman emerges victorious! Score: 456

A\*:

Path found with total cost of 54 in 0.1 seconds  
Search nodes expanded: 682  
Pacman emerges victorious! Score: 456

## Parte Bônus

### Questão 7

Não. Pois às vezes, ao ter de passar pela comida mais próxima, o agente terá que voltar um caminho já percorrido posteriormente, fazendo com que este deixe de ser o caminho mais curto. Precisamos sempre considerar o caminho passando por todas as comidas restantes antes de escolhê-lo.

### Questão 8

Sim. Utilizando a busca pelo algoritmo  $A^*$ , com a heurística adequada, vimos uma brusca mudança no número de nós expandidos; de 2448 nós expandidos pela BFS, o  $A^*$  expandiu apenas 901. Seu tempo de execução também se tornou mais rápido, enquanto o BFS teve uma execução de 0.2 segundos, o  $A^*$  fez em 0.1 segundos