

### **Busca em profundidade:**

- 1) A ordem de exploração foi de acordo com o esperado? O Pacman realmente passa por todos os estados explorados no seu caminho para o objetivo?

Sim, pois esta busca utiliza uma pilha (last-in, first-out) expandindo o nó não-expandido mais profundo, podemos observar essa expansão com mais clareza ao rodar o Pacman no mediumMaze ou openMaze e verificar a coloração do labirinto e o caminho escolhido pelo Pacman.

Podemos notar que o Pacman não passa por todos os estados explorados, uma vez que alguns becos sem saída foram explorados (verificamos pelas cores no labirinto ao rodar o programa), mas o Pacman nunca entra em tais posições.

- 2) Essa é uma solução ótima? Senão, o que a busca em profundidade está fazendo de errado?

A solução não é ótima. A busca em profundidade não faz nada de errado, porém da forma como é definida e implementada não é possível garantir que encontraremos o menor caminho até o objetivo, uma vez que a busca expande nó não-expandido mais profundo podemos encontrar um caminho, mesmo que mais longo, antes de acharmos o menor deles (como observado ao rodar o mediumMaze ou openMaze).

### **Busca em extensão:**

- 3) A busca BFS encontra a solução ótima? Por quê?

Devemos lembrar que, para este problema do Pacman, todas as ações (mover para: Norte, Sul, Leste, Oeste) possuem o mesmo custo, portanto a BFS encontra a solução ótima.

Este método de busca encontra a solução ótima, pois expande todos os estados de um nível antes de passar para o próximo. Como as ações possuem custos iguais, a solução ótima para o problema estará em um nível menor que qualquer outra solução mais longa, sendo assim o algoritmo encontrará o menor caminho entre o início e o objetivo antes de achar outro mais longo.

### **Busca de aprofundamento iterativo**

- 4) Por que a busca IDS não é ótima em busca em grafo (sem a modificação)? Por que se torna ótima adaptando a maneira de descartar nós da busca em grafo? Por que para os problemas de busca estudados neste EP não é aconselhável implementar o algoritmo IDS com busca em árvore?

A busca IDS não é ótima para busca em grafo sem a modificação, pois deixamos de expandir certos estados à medida que os exploramos. Como exploramos os estados em profundidade, o algoritmo pode não expandir um estado que leve a solução ideal porque já o excluiu em um ramo mais profundo nas buscas anteriores. Com a modificação, nós verificamos se o estado que já foi explorado não é um candidato a um caminho ótimo antes de descartá-lo, com isso, mesmo que já tenhamos atingido certo estado antes, devemos verificar se conseguimos atingi-lo com menor custo em outros ramos e expandi-lo, fazendo com que a busca IDS se torne ótima.

Para o exercício do PacMan não é ideal implementar IDS como busca em árvore, pois expandiremos muitas vezes os mesmos estados, gerando assim muitos nós e comprometendo a performance do programa (Além de sabermos que não teremos ramos infinitos, caso descartemos estados já explorados).

### Busca A\*

- 5) Você deve ter percebido que o algoritmo A\* encontra uma solução mais rapidamente que outras buscas. Por quê? Qual a razão para se implementar uma heurística consistente para sua implementação da busca A\*?

O algoritmo A\* possui um método para escolha de quais estados irá expandir primeiro (função de custo + heurística), a partir disso A\* expande os estados que mais se aproximam da resolução do problema (caso a heurística seja consistente), fazendo com que encontremos a melhor solução de maneira mais rápida.

Como a expansão dos estados depende da heurística, caso ela seja definida de forma inconsistente a busca A\* pode se tornar muito ineficiente, uma vez que buscará expandir os estados que não estão próximos da solução do problema, muitas vezes levando a soluções mais longas.

- 6) O que acontece em openMaze para as várias estratégias de busca?

DFS – acaba encontrando um caminho muito longo devido a ordem de exploração dos nós e a ordem de pegar os sucessores de um estado (o PacMan vai bastante para a esquerda, depois direita, esquerda), porém era o esperado, pois a busca expande um ramo ao máximo.

BFS – expande muitos nós como era o esperado, mas encontra uma solução ótima para o problema.

IDS – com a modificação encontra um caminho ótimo, porém demora bastante para encontrá-lo já que expande várias vezes estados que não levarão à solução.

A\* – encontra a solução ótima expandindo menos estados, uma vez que é direcionada pela heurística.