**Lab 3: Informed Search in Pac-Man**

Name: Nguyễn Đình Khánh Ngân

ID: ITCSIU22236

1. **Implement the Best-First Search (BFS) algorithm in the bestFirstSearch function in search.py. Test your code the same way you did for other search algorithms**

Best First Search function

**A screen shot of a computer code

AI-generated content may be incorrect.**

Implement function in different maze layout

tinyMaze

**A screen shot of a computer program

AI-generated content may be incorrect.**

mediumMaze

**A screen shot of a computer

AI-generated content may be incorrect.**

bigMaze

**A screen shot of a computer

AI-generated content may be incorrect.**

**Does BFS find a least cost solution? How many nodes are expanded?**

Yes, BFS find a least cost solution

tinyMaze: 15 nodes

mediumMaze: 269 nodes

bigMaze: 620 nodes

1. **Implement the A\* Search algorithm in the aStarSearch function in search.py. Use the same algorithm as shown in your text (or class). aStarSearch function takes an optional heuristic function as an argument. The heuristic function itself takes two arguments (a state in the search problem, and the problem itself). search.py provides a nullHeuristic function that you can look at. Also, in the searchAgents.py a Manhattan heuristic as well as Euclidian heuristic function is defined. Test your code the same way you did for other search algorithms**

A\* Search function

A screen shot of a computer program

AI-generated content may be incorrect.

Implement A\* search function (search.py provides a nullHeuristic)

tinyMaze

A screen shot of a computer program

AI-generated content may be incorrect.

mediumMaze

A screen shot of a computer

AI-generated content may be incorrect.

bigMaze

A screen shot of a computer

AI-generated content may be incorrect.

Implement A\* search function (searchAgents.py a Manhattan heuristic)

A screen shot of a computer

AI-generated content may be incorrect.

1. **Compare table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Best First Search** | | | **A\* Search** | | |
| **Maze** | Nodes explored | Solution length | Is it optional | Nodes explored | Solution length | Is it optional |
| **Tiny** | 15 | 8 | Yes | 15 | 8 | Yes |
| **Medium** | 269 | 68 | Yes | 269 | 68 | Yes |
| **Big** | 620 | 210 | Yes | 620 | 210 | Yes |

* What happens on openMaze for the various search strategies?

BreathFirstSearch, AStarSearch, UniformCostSearch, and BestFirstSearch all display the same path and path cost

Despite not optimizing, depthFirstSearch searches every inch of an open maze, increasing the path cost

* Based on the above, a short discussion/reflection of how the searches compare to each We can observe that in three areas—nodes expanded, solution length, and optimal—BestFirstSearch and A\*Search yield identical results. They are all conducted in small, medium, and large mazes, which also makes the testing process more transparent.