**VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY**

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Ảnh có chứa vòng tròn, biểu tượng, Nhãn hiệu, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

**ARTIFICIAL INTELLIGENCE**

**IT159IU**

**REPORT LAB 3**

**Instructor:**

**Dr. Nguyen Trung Ky**

**Dr. Ly Tu Nga**

**Nguyen Huynh Ngan Anh - ITDSIU23003**

1. **Setup:**
2. **Running befs, astar in tinyMaze, mediumMaze, bigMaze with nullHeuristic and manhattanHeuristic:**

bestFirstSearch algorithm

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

With **nullHeuristic**:

* Befs:

Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

* A\*:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

With **manhattanHeuristic**:

* Befs:

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

* A\*:

Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

1. **Running befs, astar in openMaze with nullHeuristic and manhattanHeuristic:**

With **nullHeuristic**:

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

With **manhattanHeuristic**:

**Ảnh có chứa ảnh chụp màn hình, văn bản

Nội dung do AI tạo ra có thể không chính xác.**

1. **Exercise 1:**

With **nullHeuristic**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Best First Search** | | | **A\* Search** | | |
| **Maze** | **#nodes expanded** | **Solution length** | **Is it optimal?** | **#nodes expanded** | **Solution length** | **Is it optimal?** |
| **tiny** | 15 | 8 | No | 15 | 8 | No |
| **medium** | 269 | 68 | No | 269 | 68 | No |
| **big** | 620 | 210 | No | 620 | 210 | No |

With **manhattanHeuristic**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Best First Search** | | | **A\* Search** | | |
| **Maze** | **#nodes expanded** | **Solution length** | **Is it optimal?** | **#nodes expanded** | **Solution length** | **Is it optimal?** |
| **tiny** | 8 | 8 | Yes | 14 | 8 | No |
| **medium** | 78 | 74 | Yes | 221 | 68 | No |
| **big** | 466 | 210 | Yes | 549 | 210 | No |

1. **Exercise 2:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Best First Search** | | | **A\* Search** | | |
| **Maze** | **#nodes expanded** | **Solution length** | **Is it optimal?** | **#nodes expanded** | **Solution length** | **Is it optimal?** |
| **nullHeuristic** | 682 | 54 | No | 682 | 54 | No |
| **manhattanHeuristic** | 89 | 68 | Yes | 535 | 54 | No |

Overall, the performance of each search strategy on OpenMaze depends on the characteristics of the maze and the search problem, as well as the parameters and implementation details of the search algorithm.

* For Null Heuristic, BEFS and A\* behave like UCS, they expand many nodes and do not necessarily find the best path.
* For Manhattan Heuristic, BEFS is faster, A\* is more optimal.

=> BEFS with Manhattan Heuristic expands fewer nodes but has a longer path 🡪 use for efficiency.

=> A\* with Manhattan Heuristic finds a shorter path but expands more nodes 🡪 use for optimal paths.

1. **Exercise 3:**

**nullHeuristic:** is straightforward but ineffective because it consistently returns 0, regardless of where the objective or the Pac-Man is. It lacks knowledge of the area. In other words, a null heuristic does not provide any information or guidance to the search algorithm and is equivalent to performing a search algorithm.

**manhattanHeuristic:** is more intelligent because it makes calculations based on the locations of the Pacman and the objective. It gives the algorithms clues regarding how near or far Pacman is from the objective. When a heuristic function is used in a search algorithm, it is usually incorporated into the cost function or evaluation function that determines the priority of nodes in the search frontier. For example, in A\* search, the cost of a node is the sum of the actual cost of the path so far and the estimated cost to the goal based on the heuristic function. The evaluation function for A\* is the sum of the cost function and a tie-breaking function that ensures nodes with the same cost are explored in a consistent order. It is knowledgeable about the area.

1. **Exercise 4:**

With ignorant algorithms in Assignment#2, BEFS and A\* both perform equally well if combined with the nullHeuristic.

But when combined with ManhattanHeuristic, BEFS, and A\* outperform the same uninformed algorithms in Assignment#2 while using fewer enlarged nodes. In my trials, BEFS outperforms A\*.