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

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

Logo

Description automatically generated

**INTRODUCTION TO ARTIFICIAL INTELLIGENCE**

Course by Dr. Nguyen Trung Ky

**LAB #3:**

**INFORMED SEARCH IN PAC-MAN**

Student's Name: Nguyễn Đức Nguyên Phúc

Student's ID: ITDSIU22108

**Exercise 1:**

With **nullHeuristic**:

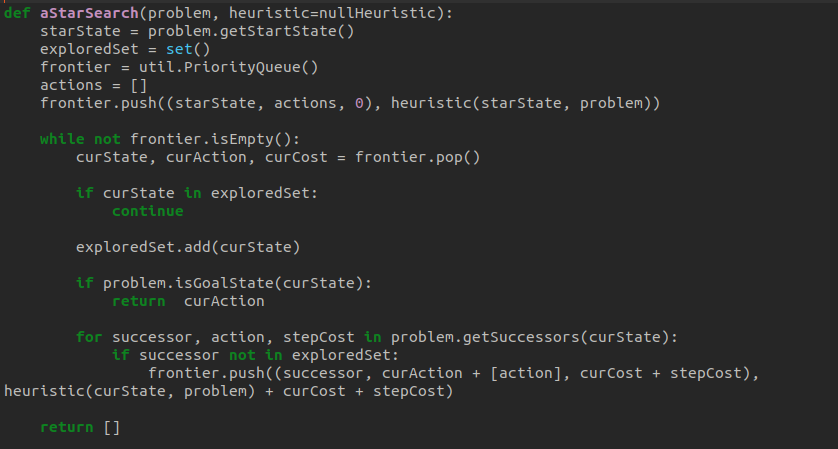
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Best First Search** | | | **A\* Search** | | |
| **Maze** | **#nodes expanded** | **Solution length** | **Is it optimal?** | **#nodes expanded** | **Solution length** | **Is it optimal?** |
| **tiny** | 15 | 8 | No | 15 | 8 | No |
| **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** | 9 | 8 | Yes | 15 | 8 | No |
| **medium** | 80 | 74 | Yes | 227 | 68 | No |
| **big** | 480 | 210 | Yes | 556 | 210 | No |

**Exercise 2:**

DFS method function as intended.  
  
There is a new problem with my version of the BFS, UFS, BEFS, and A\* algorithms. GameState is causing speed problems since the main data structures, Queues and PriorityQueues, hold a lot of redundant data.  
Therefore, in order to decide whether or not the currentState is investigated, I added the following condition:



After that, the algorithms work flawlessly and stop performing pointless tests.

All things considered, the effectiveness of every search method on OpenMaze is contingent upon the features of the labyrinth and the search problem, in addition to the search algorithm's parameters and implementation specifics.

=> DFS takes the longest path but reaches the answer quickly.

=> Although they have the shortest path, BFS, UFS, BEFS, and A\* locate the solution slowly.

**Exercise 3:**

NullHeuristic: is simple but inefficient because it always returns zero, regardless of the Pac-Man's or the objective's location. It is ignorant about the field. Put differently, a null heuristic is the same as running a search algorithm; it offers no more information or direction.

**manhattanHeuristic:** is more sophisticated since it performs computations depending on Pacman's and the goal's locations. It provides hints to the algorithms about Pacman's distance from the goal. A heuristic function is typically included into the evaluation function or cost function of a search algorithm, which establishes the nodes' precedence in the search frontier. For instance, in an A\* search, a node's cost is the total of its actual path costs thus far plus its anticipated goal-to-cost estimate from the heuristic function. The cost function with a tie-breaking function that guarantees nodes with the same cost are investigated in a consistent order make up the evaluation function for A\*. It knows a lot about the field.

**Exercise 4:**

When paired with the nullHeuristic, BFS and A\* perform equally well with the uninformed algorithms in Assignment #2.   
However, BFS and A\* work better together with ManhattanHeuristic than they do alone, utilizing fewer expanded nodes, and beating the same ignorant algorithms in Assignment #2. In my tests, BFS performs better than A\*.