Full name	Nguyễn Việt Kim
ID	**21127333**
Class	21CLC08

Lab01: Search Strategies

REPORT

A. Check list

No.	Task	Progress
1	Implement all of the searh strategies: - *Breadth-first search and Depth-first search* - *Uniform-cost search* - *Greedy-best first search* using the Manhattan distance as heuristic - *Graph-search A** using the same heuristic as above	50%
2	For each search strategy, print to the console the following information: - (The time to escape the maze) - The list of explored nodes (in correct order) - The list of nodes on the path found (in correct order).	100%
3	Create some example mazes as input text files in the INPUT folder and corresponding search results to them as output text files in the OUTPUT folder.	50%

B. Brief description of main functions

The usage, the input parameters, and the returned result of each function are noted carefully as comments in my source code. You can read the below brief description along with those comments for better understanding.

1. ***Main (main.py)***

In *'main.py'*, the input text file of the maze is read first and stored into a *dictionary* data structure, each one represents the cell and the data that the cell has, including the nearby cell and the cost to that cell.

By default, the input text file's path is *"maze.txt"*. Since there's a shortage in my time, only the given maze is used as a test case.

2. ***Input/Output handle (IOHandle.py)***

'file_handle.py' is created in order to implement some functions related to Input/Output Handle such as:

- Reading the input file text of a maze *(inputMaze)*

3. ***Breadth-first search (BFS.py) – Graph search***

The *breadth-first search* algorithm (BFS for short) is implemented in *'BFS.py'* with the main function is named *BFS*. This function returns 5 values: (*the maze, the list of explored nodes, the list of nodes on the path found, the start cell and the goal cell).*

4. ***Uniform-cost search***

5. ***Greedy-best first search***

6. ***A* search (A star.py)***

The function *h()* is used to calculate the Manhattan distance (heuristic) of a cell.

The function *g()* is used to return the distance from the current cell to the neighbour cell.

The function *A_star* is used as the main function for A* algorithm.

THE END