Group Members:

1. Qui, Christian Michael
2. Regencia, Josiah Eleazar

Requirements:

1. Python2.7.* or Python3

The project is an implementation of uninformed and informed search algorithms using a self-defined PEAS. The objective of this simulation is that in a 3 x 3 chess board, where a knight chess piece and an immobile enemy piece are randomly generated anywhere in the board other than [1,1] (because this is the center of the board, and it's impossible for the knight to visit or move from it.), the knight must search for the enemy piece. The knight piece must follow the traversal rules in the game of chess, where it must traverse 2 cells horizontally/vertically, and move an additional cell horizontally or vertically, depending on the former movement. The PEAS are as follows:

Performance measure – The knight reaches the cell where the enemy is located

Environment – 3 x 3 chessboard.

Actuator – "L" traversal of the knight.

Sensor – Knight knows its location, and its next moves.

Our informed search method is the Greedy Best-First Search.

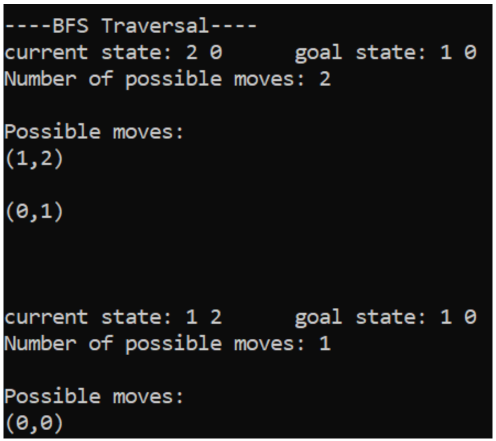
Notes on how to run the Project:

1. Main file is knight_tour.py
2. In contrast to our project 1, where it accepts an additional parameter such as BFS and DFS, this program runs all algorithms (BFS, DFS, Greedy Best First Search)

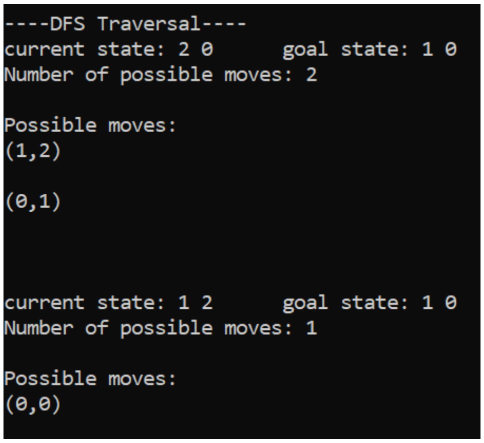
To run, simply type python knight_tour.py in the command line/terminal. This will run all algorithms for your convenience. 👌

Notes on the output:

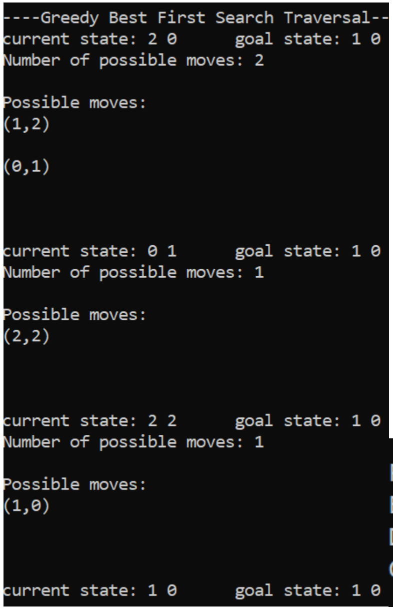
1. Below is a sample output:



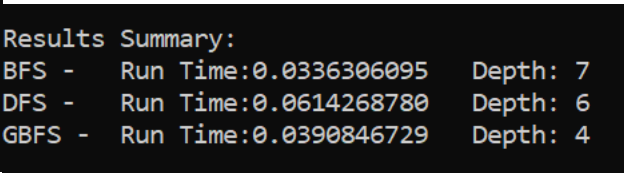
```
----BFS Traversal----
current state: 2 0      goal state: 1 0
Number of possible moves: 2

Possible moves:
(1,2)

(0,1)



current state: 1 2      goal state: 1 0
Number of possible moves: 1

Possible moves:
(0,0)
```



```
----DFS Traversal----
current state: 2 0      goal state: 1 0
Number of possible moves: 2

Possible moves:
(1,2)

(0,1)



current state: 1 2      goal state: 1 0
Number of possible moves: 1

Possible moves:
(0,0)
```



```
----Greedy Best First Search Traversal--
current state: 2 0      goal state: 1 0
Number of possible moves: 2

Possible moves:
(1,2)

(0,1)



current state: 0 1      goal state: 1 0
Number of possible moves: 1

Possible moves:
(2,2)



current state: 2 2      goal state: 1 0
Number of possible moves: 1

Possible moves:
(1,0)



current state: 1 0      goal state: 1 0
```



```
Results Summary:
BFS -    Run Time:0.0336306095   Depth: 7
DFS -    Run Time:0.0614268780   Depth: 6
GBFS -   Run Time:0.0390846729   Depth: 4
```

2. The output has 4 parts – BFS traversal, DFS traversal, GBFS traversal, and the results summary
3. In the first two lines, the randomly generated locations of the knight and the enemy are printed.
   - All algorithms uses the same randomly generated coordinates
4. The lines that follow this are the traversals of the knight.
5. When the goal has been reached for all 3 algorithms, the summary is printed
   - Run time is printed
   - Farthest depth level is also printed