cmpe480 - 2024 Fall - HW1

Submit two unzipped files: name-lastname.pdf and name-lastname.py

You are given an NxN chessboard. Your agent moves the white pieces and might have a bishop (♗) and/or rook (♖) and/or knight ♘ on the board. Your opponent has between 1 and 9 pawns (♟). Only your agent plays. Each action corresponds to selecting a piece and making a possible move with that piece. The goal is to capture all the pawns.

Action costs are as follows (who knows why..)

- bishop (♗): 10
- rook (♖): 8
- knight (♘): 6

An example board is where you have a bishop, rook and knight and the opponent has four pawns. There are also several obstacles (denoted by x)

```
. . . B R K . . .
. . x x . . x . .
. . . . . . . . .
. . x 1 . 2 x . .
. . . . . . . . .
. . . . . 3 . . .
. . 4 . . . . . .
```

Another example board is as follows where you have a rook and the opponent has 6 pawns:

```
. R . . .
. 1 x . .
. 2 x . .
. 3 x . .
. 4 5 6 .
```

You need to implement Uniform Cost Search (UCS), Greedy Search (GS) and A* Search (AS). Two heuristics will be used for GS/AS:

h1:

- If any pawn at the same row or column with your rook: # of pawns x 8
- If no pawn at the same row or column with your rook: (# of pawns + 1) x 8

This heuristic will be evaluated in boards that only contain rook.

h2= Your heuristic function. This heuristic will be evaluated with any possible board.

You will report

- The number of expanded nodes.
- The cost of the solution.
- The solution.

Tie conditions: The nodes in the fringe might have the same values. In such cases, follow the following rules to decide which node to remove first:

1- When there is a tie in removing nodes from the fringe, prioritize the nodes inserted earlier (from the successor list).
2- When there is a tie in removing nodes from the fringe for moving different pieces, use the following precedence rules:

- First, knight (♘), then bishop (♗), and finally rook (♖) are removed.

3a- When there is a tie in removing nodes from the fringe for the knight (♘), use the following precedence:

	4		5	
3				6
		♘		
2				7
	1		8	

3b- When there is a tie in removing nodes from the fringe for the bishop (♗), use the following precedence rules:

- Order: South-West, North-West, North-East, South-East
- Given the same direction, smaller moves have higher precedence. For example:

4				6
	3		5	
		♗		
	1		7	
2				8

3c- When there is a tie in removing nodes from the fringe for the rook (♖), use the following precedence rules:

- Order: South, West, North, East
- Given the same direction, smaller moves have higher precedence. For example:

		6		
		5		
4	3	♖	7	8
		1		
		2		

Running your code:

python3.10.9 name-lastname.py board.txt output.txt UCS/GS/AS h1/h2

Do your best to find a good heuristic for h2. You must describe your h2 in a pdf file and explain why it is admissible and complete.

The board.txt examples are given above.
Output.txt should be in the following format:

```
expanded: <number of expanded nodes>
path-cost: <path case of the solution>
h1: <h1 value of the initial board>
h2: <h2 value of the initial board>
```

The following lines: step-by-step solution as follows:

Example solution:

```
. R . . .
. 1 x . .
. 2 x . .
. 3 x . .
. 4 5 6 .
*********
```

```
. . . . .
. R x . .
. 2 x . .
. 3 x . .
. 4 5 6 .
*********
. . . . .
. . x . .
. R x . .
. 3 x . .
. 4 5 6 .
*********
. . . . .
. . x . .
. . x . .
. R x . .
. 4 5 6 .
*********
. . . . .
. . x . .
. . x . .
. . x . .
. R 5 6 .
*********
. . . . .
. . x . .
. . x . .
. . x . .
. . R 6 .
*********
. . . . .
. . x . .
. . x . .
. . x . .
. . . R .
*********
```

Important: I want to see an almost exact code segment for the GRAPH-SEARCH algorithm. Name the corresponding functions similarly so that I can search and find them easily.