**ITCS 3153 – Intro to Artificial Intelligence – Spring 2023**

# Assignment 3: Implementation
# Uninformed & Informed Search

## Recommended Steps & Hints

### Step: Run the Jupyter notebook

Run the jupyter notebook of the assignment “Assignment3.ipynb” until an error is returned to get an idea about the problem. The errors are due to the fact that not all functions we need are implemented in search.py and maze.py.

### Step: Specify search problem

Read and understand the code of the classes “SearchProblem”, “Node” and “MazeProblem”. Once you have done this, complete the code of the methods “actions” and “result” of the class “MazeProblem” in maze.py that specify the list of applicable actions and the transition model for applying an action for the maze grid problem we want to solve.

### Step: Implement expand method

Implement the “expand” method in search.py as a building block of the search algorithms, cf. Section 3.3 of the textbook.

### Step: Implement search methods (Part I)

We are now ready to implement the search algorithms. Begin with the functions “depth_first_search” and “breadth_first_search” in search.py, and implement them while recalling from Section 3.3 in the textbook how they work.

HINT: You can define and additional function best_first_search in search.py that implements best-first search and use an appropriate evaluation function to save some lines of code.

### Step: Heuristic function

For the informed search methods, implement the code for the computation of the Manhattan distance of a state to the goal state in the method “h” of the class “MazeProblem”.

## Step: Implement search methods (Part II)

Continue with the implementation of the functions “greedy_first_search” and “astar_search” in search.py to also get the informed search algorithms working.

## Step: Test code

Check whether your code works by executing the respective cells in the jupyter notebook of the assignment “Assignment3.ipynb”. Is the behavior of the different search methods as you would expect?

It is highly likely that the number of reached nodes and/or expanded nodes is not correctly returned by the “print” lines (unless you already worked on this at this point).

## Step: Implement tracking of number of reached nodes and expanded nodes

Return to the functions of search.py and maze.py and check how the variables _reached, _reachedlist and _expanded are set when initializing an object of the class MazeProblem. Figure out what needs to be changed in the functions of search.py to make sure that these variables are kept up to date throughout the search algorithm.

## Step: Test code again

Check whether running the cells of the jupyter notebook now returns the correct numbers for nr. of expanded nodes, and whether all the paths are returned correctly, etc. If not, go back and iterate on the code.

## Step: Answer Question of 6. of Assignment Requirements

Do this in the jupyter notebook “Assignment3.ipynb”.

## (Not obligatory): Work on Bonus Question.

**Rough Grading Rubric:**

| | |
|---|---|
| Specification of Search Problem | 4 |
| Expand Function | 2 |
| Correctness of Breadth-First Search | 3 |
| Correctness of Depth-First Search | 3 |
| Correctness of Greedy Best-First Search | 3 |
| Correctness of A* Search | 3 |
| Correct tracking of path length & expanded nodes | 4 |
| Question 6 | 3 |
| Bonus Question | Additional 4 |