

## COMP 5600/6600

### Assignment 1 – Using Search for Robot Path Planning

**Due:** 06/20/2025 11:59 pm

**Submission:** On Canvas

#### Problem Scenario

You are designing a robot whose goal is to move from one location to another, just as we discussed in Lecture 6. You want to evaluate the ability of informed and uninformed searches to find the optimal path (shortest) to get from the start to the goal state. To help you, I have converted the map of the building into a state-space graph. Your goal is to implement 2 uninformed search algorithms (BFS and DFS) and 1 informed search (A\* algorithm) to search for an optimal path from the start node to the end node. You will be given 3 test cases on which to evaluate your implementations.

#### Deliverables:

You will need to have your own implementation of each of the three algorithms as well as a short report that discusses your design choices, your implementation strategy, and a comparison of the performance of the three algorithms on each of the test cases. You can compare them using any metric(s) of your choice, such as success rate, time taken to find a solution, and number of steps taken to find a solution. You should briefly describe your findings and provide your insights on which is suitable for this problem. **For the informed search (A\* algorithm) you must devise and evaluate at least 2 heuristics of your choice.** You are free to choose any function as your heuristic. In your report, you should describe why you chose it, whether it is an admissible heuristic, and whether it helped the A\* algorithm perform better than uninformed search.

Here are some more details about your assignment.

- **Input:** 3 test cases, with varying levels of complexity, are provided. Each test case consists of 2 files. The first, labeled “TestCase\_XX\_EdgeList.txt” is a text file where each line corresponds to an edge list of the form  $\langle n1, n2, w \rangle$ , which indicates an edge between nodes  $n1$  and  $n2$  with a weight of  $w$ . The second file, labeled “TestCase\_XX\_NodeID.csv”, is a CSV file with each line of the form “ $n1,x,y$ ”, where  $x$  and  $y$  are the coordinates of the state  $n1$ .

- **Expected Output:** Your program should print out a list of states visited by your algorithm, from the start state (indicated by the first line of the NodeID file) to the goal state (the last line of the NodeID file).
- **Submission Format:** Your code must be an IPython Notebook. You can have text blocks to write your report and the code blocks for your implementation. You can use [Google Colab](#) for implementing your code since your code will be evaluated on Colab so that everyone's code is evaluated on a standard platform. You can download your file for submission by going to "File->"Download"->"Download .ipynb".

To Verify your implementation, the expected output of BFS and DFS are given below. Since heuristic functions can vary and hence result in different solutions, the output for A\* is not provided here.

Case 1:

-----

BFS: ['N\_0', 'N\_1', 'N\_6', 'N\_2', 'N\_5', 'N\_7', 'N\_3', 'N\_10', 'N\_12', 'N\_15', 'N\_11', 'N\_13', 'N\_17', 'N\_20', 'N\_16', 'N\_8', 'N\_14', 'N\_18', 'N\_22', 'N\_21', 'N\_9', 'N\_19', 'N\_23', 'N\_4', 'N\_24']

DFS: ['N\_0', 'N\_1', 'N\_2', 'N\_3', 'N\_6', 'N\_7', 'N\_12', 'N\_17', 'N\_22', 'N\_23', 'N\_13', 'N\_18', 'N\_19', 'N\_24']