

# CSC 410 A-Star Assignment

Zac Swisher

## Useful prompts:

To get the code, I first prompted ChatGPT; “Give me the pseudocode to do the a star algo to find the shortest path from start node to end node.”

Then, I prompted Claude Code with “Using this pseudocode, code the algorithm in python with regards to the following input structure (copied and pasted from the assignment PDF):

The input text file has the following components: ● The first line of the file has a single integer that represents the number of cities (i.e., the vertices in the graph. Call this number N. ● The next N lines will each have a label and a pair of numbers. The label will be a string of letters (with no spaces or non-alphabetic characters). The two integers should be interpreted as the X and Y coordinates (respectively) of a city. The three components (label, x-coordinate, y-coordinate) will be separated by commas. NOTE: The label that is provided first in this section is the starting city for the A\* search, and the city that is listed last in this section is the target destination for the A\* search. ● The remaining lines should have pairs of labels (separated by commas) that represent the edges of the graph. The graph is symmetric and so edges can be traveled in both directions. This is to say that the following two lines are equivalent A,B B,A Either of these lines in the file would indicate the existence of an edge between A and B. Print out both the shortest path, as well as the order in which the A-Star algorithm explored the vertices in the graph.”

The algorithm worked and I did not need to modify anything. However, I did have to modify the input parsing. I also modified the output formatting for aesthetic purposes. There was no code that I had to write entirely myself. Lastly, I prompted ChatGPT; “give me the steps to do the a star algo to find the shortest path from start node to end node. im working it out on paper.” Then I graphed example1 on paper and verified the results.

## Summary

My experience working on this project using AI tools closely resembles my previous personal experience with using AI to assist with similar coding tasks. I found that the tasks that the AI was good at was the algorithm portion of the code. ChatGPT was able to

generate pseudocode and Claude Code was able to convert those mechanical steps into working python code. However, the AI tools struggled with more open-ended tasks that I didn't provide much structure or instructions for. For example, the output was cluttered and did not display the order in which the A-Star algorithm explored the vertices in the graph. This experience portrays the current state of AI tools for software development well. Since AI was able to code this algorithm, I believe future Junior/New Grad Software Engineers will be overly reliant on AI. If employers don't embrace this, they will not see new grads as qualified. This might result in junior engineers who aren't reliant on AI to be more valuable in the near future.