

CSCI 4511

## Project 1

GitHub Link: [https://github.com/Hermanubis/CSCI4511\\_Project1](https://github.com/Hermanubis/CSCI4511_Project1)

- For the uninformed search algorithm, I choose to implement Dijkstra's algorithm to find the shortest path. Dijkstra's algorithm is uninformed since it doesn't require any information about the distance from current state to the goal, thus it also doesn't need to calculate of any heuristic value. So for each vertex, only the distance to the start vertex is considered, not its distance to the goal.

- For the informed search algorithm, A\* algorithm is implemented. It is very similar to Dijkstra's algorithm except it uses the heuristic values for each vertex, which is the estimated distance from the current vertex to the goal. So the algorithm always prioritize searching the node that has the smallest estimated total distance by adding the distance from the start vertex and the estimated heuristic value. For my program, I choose the heuristic to be the average distance from the current vertex's square to the square of the goal vertex.

- Both algorithms are implemented using python and assumes the graph data is named 'p1\_graph.txt' and in the same directory.

- The execution time for each algorithm is calculated. I tested different combinations of source and destination vertices, and on average the execution time of the two algorithms is very similar. The A\* informed search algorithm is not consistently faster than Dijkstra's algorithm. I suspect this is because the size of the dataset given is relatively small, on a larger data set with more edges, the efficiency advantage of informed search will be much more obvious.