**Artificial Intelligence – *Project 1 Report***

Qinyang Li G33129490

**Algorithm Choice**

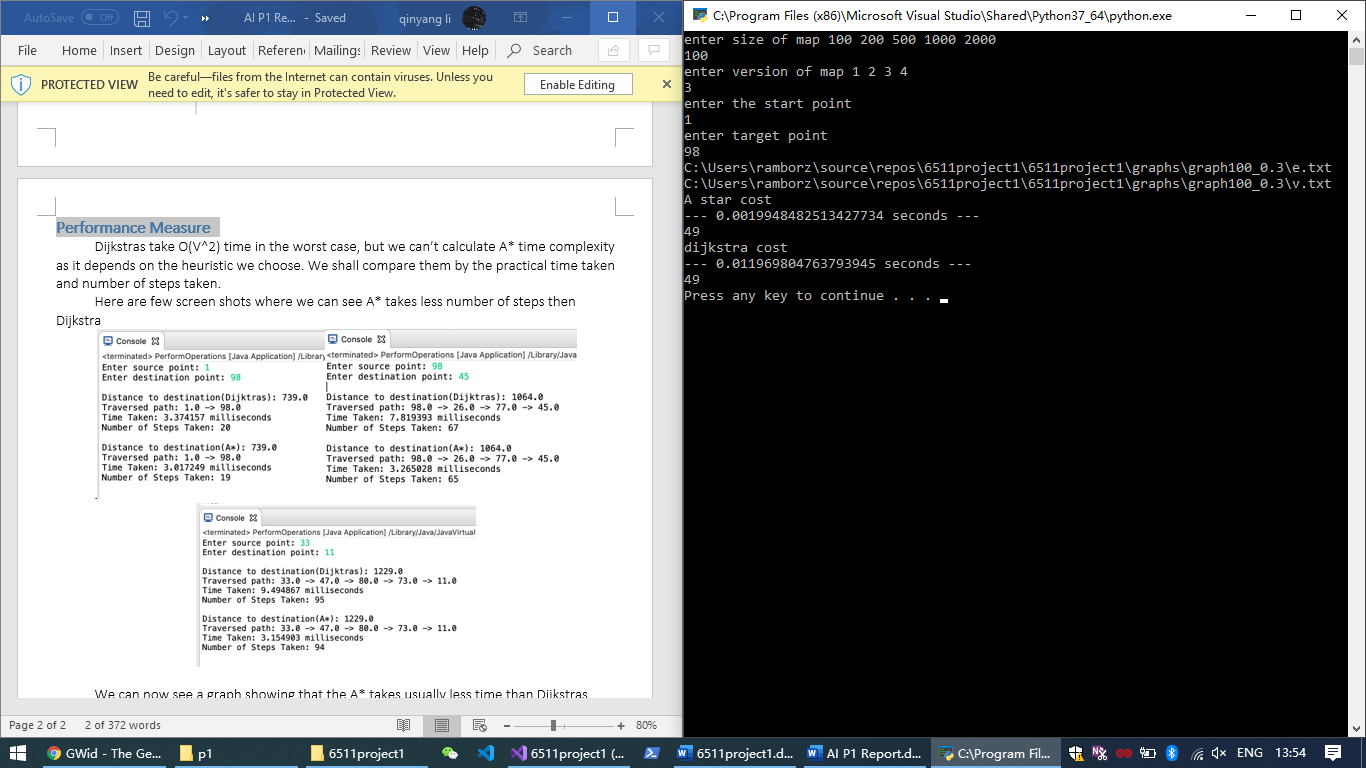
In this project, I used Python implemented a Dijkstra uninformed search and A star informed search. Treat every line in the e.txt as a node point and v.txt as edge between 2 nodes. Using pseudocode available at Wikipedia, Dijkstra algorithm was straight forward. For A star, I first implemented a uniform cost search algorithm. By adding the Euclidean distance between the next node and end node to edge cost from this node to next node F(n)=g(n)+h(n), It will become the value that influence the position of this node in the priority queue.

**Problem**

We are given a file which contains vertices and edges. We need to read those files and add them to our graph. After reading all the data in a particular file, we apply our Search Algorithms and print desired outputs and performance measures.

**How to run script**

The program first asks to input the map size and version to determine which files are used. Second, it will ask the start and target point. After storing info into workable data structure, It will time Dijkstra and A star search separately. Finally, it will show the cost of each search.



**Performance Measure**

The Dijkstra and A star algorithm performed similar with small data set. However, with larger data set, A star algorithm will be noticeably faster than the Dijkstra.