**Controls**

1. Map configuration i.e sizes should be the same and the same amount of obstacles should be placed for each user
2. Same heuristic function they should be the same which they are
3. Same termination conditions i.e after a certain amount of time and iterations done
4. Implementation details – use the same data structures i.e vectors and priority queues etc
5. Statistical analysis to evaluate speed
6. Randomization – random start and end positions for testing to ensure no bias
7. Number of trials for the data
8. Path length – static start and end pos

**Comparisons**

1. Performance- speed of the algorithm and the memory usage
2. Optimality – compare how often each algorithm returns to the best path
3. Robustness – speed with obstacles on the path
4. Scalability – for increase and decrease of the grid sizes
5. Implementation – how difficult each algorithm is to implement
6. Depth of code i.e how many for nested for loops or conditionals that increase the complexity of the code

Visual studio – profiler

Blue line is D\*

Orange line is A\*

Grey line is IDA\*

The graph will work as follows the category is the length of the path i.e 10-1000 cells long etc

The left side of the graph is the time taken for the algorithm to finish the path

There will be a total of 4 graphs for space 2500 cells

Up to 4 walls placed i.e 4 corrections needed for Dstar

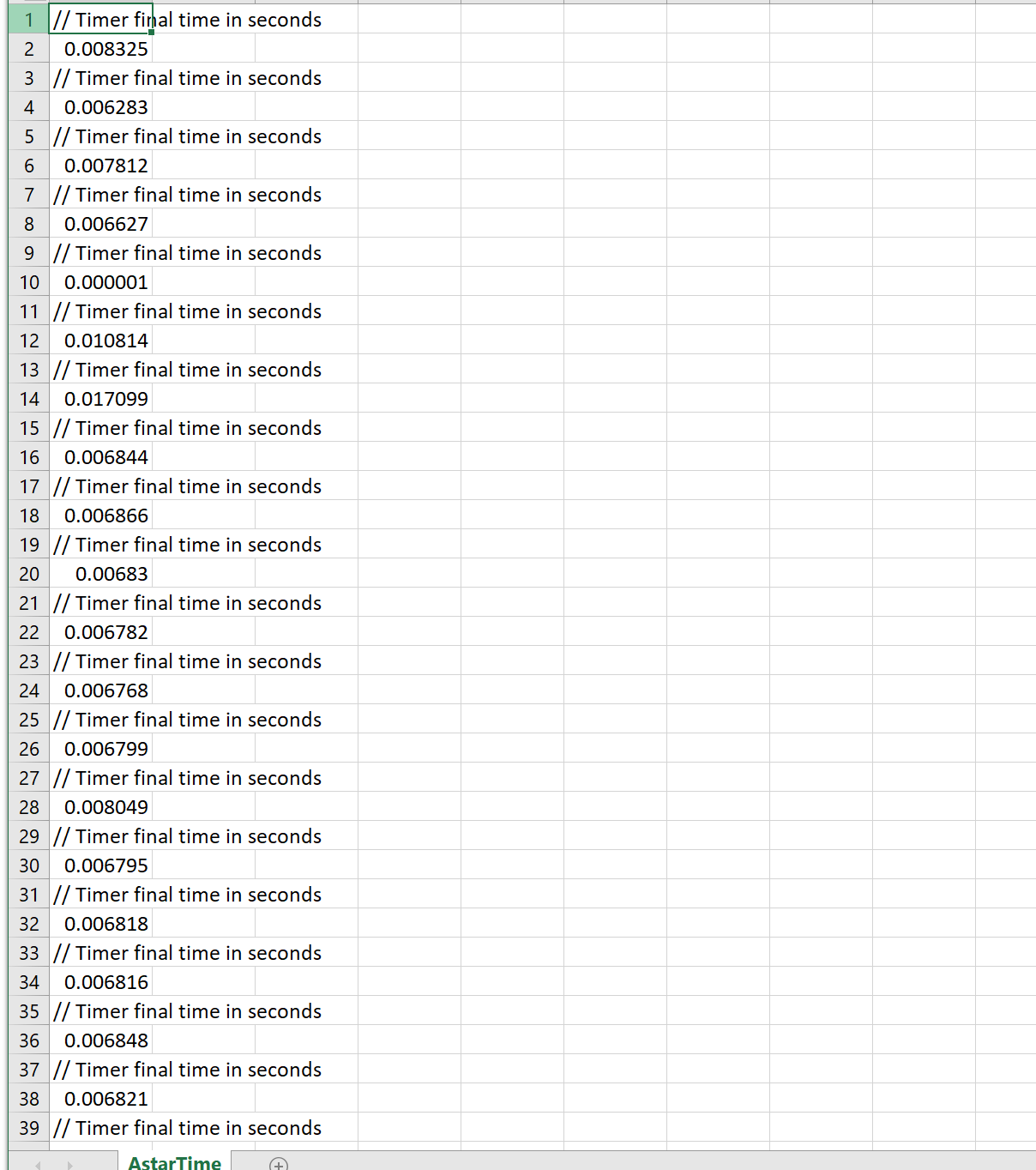
There will be a total of 4 graphs for space 6400 cells

Up to 4 walls placed i.e 4 corrections needed for Dstar

There will be a total of 4 graphs for space 22000 cells

Up to 4 walls placed i.e 4 corrections needed for Dstar

Example of astar data collected on a now wall path of varying lengths on the 2400 celled grid



For the graph the average time will be collected

I will run the algorithm up for 40 times and get the average off of that and use that inside of the graphs the actual data i.e non average time will be displayed underneath each graph in a table for each algorithm in a table under each condition