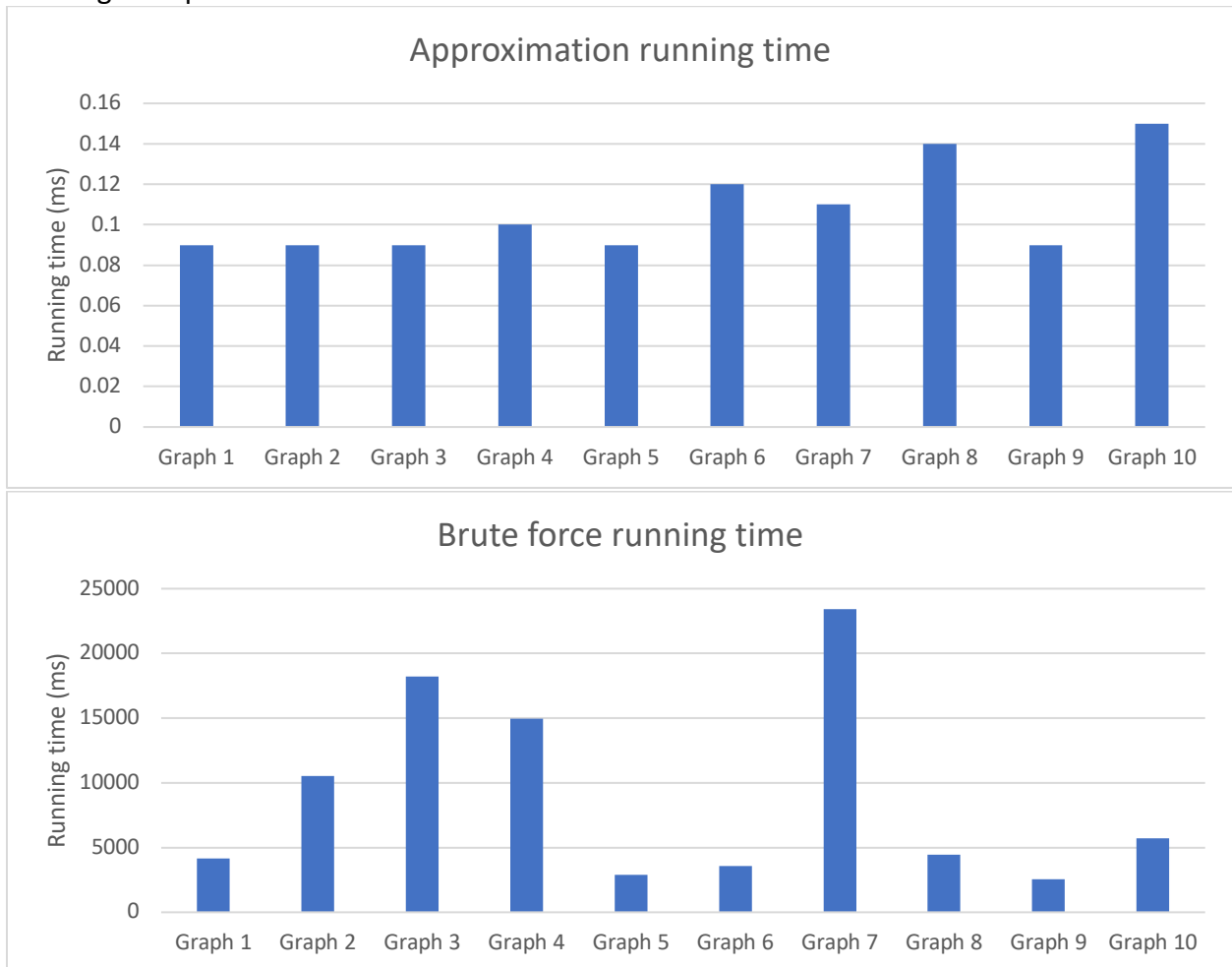Project 6 report:

Running time plot:

Approximation running time

Running time (ms)

| | Graph 1 | Graph 2 | Graph 3 | Graph 4 | Graph 5 | Graph 6 | Graph 7 | Graph 8 | Graph 9 | Graph 10 |

0.16, 0.14, 0.12, 0.1, 0.08, 0.06, 0.04, 0.02, 0

Brute force running time

Running time (ms)

| | Graph 1 | Graph 2 | Graph 3 | Graph 4 | Graph 5 | Graph 6 | Graph 7 | Graph 8 | Graph 9 | Graph 10 |

25000, 20000, 15000, 10000, 5000, 0

Distance calculated table:

Graph	Brute force distance	Approximation distance	Approx / BF
Graph 1	344.71	625.4	1.81427867
Graph 2	357.99	708.7	1.97966424
Graph 3	344.69	562.5	1.63190113
Graph 4	341.64	573.19	1.67776022
Graph 5	331.76	583.54	1.75892211
Graph 6	263.28	791.93	3.00793832
Graph 7	361.07	617.52	1.71025009
Graph 8	296.01	481.39	1.62626263
Graph 9	331.66	625.24	1.88518362
Graph 10	325.99	519.79	1.59449676

The graph with the worst ratio are graph 6 and graph 2:
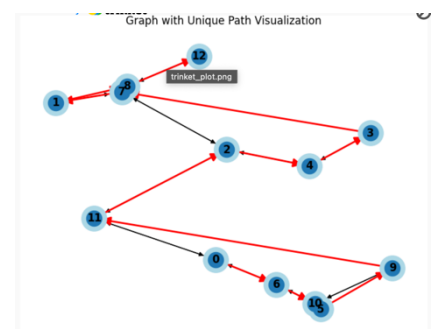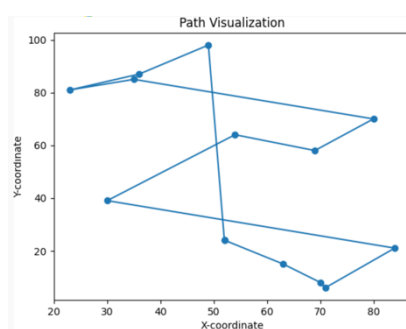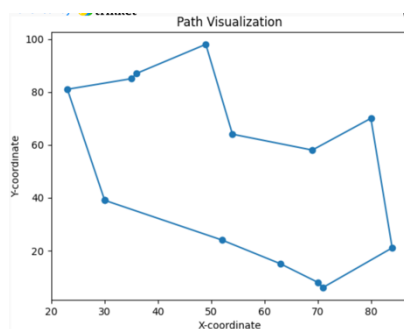
Graph 6 true shortest path, aprroximation path and MST (ignore black path)
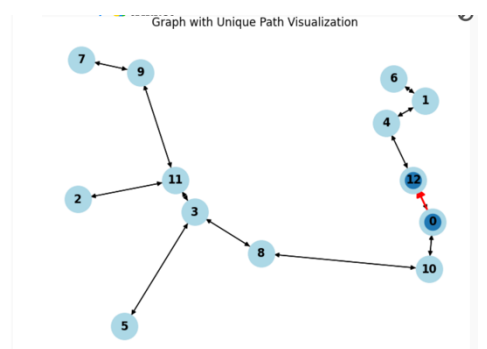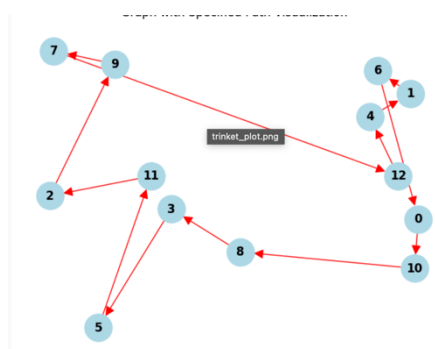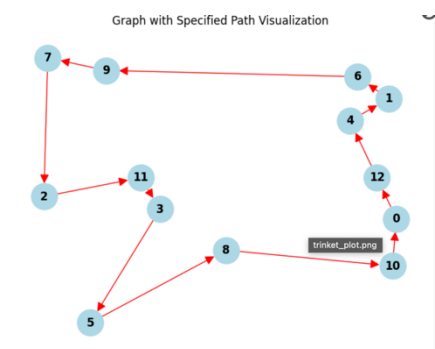
Graph 2 true shortest path, approximated shortest path and MST

The approximation is not totally ideal, as the approximation is around 1.7 times longer than the actual shortest path. What makes a graph a good approximation is that the path that is not

within the MST is short. To optimize brute force, I return the recursion when the current distance is longer the the current calculated minimum distance, and returns immediately if it has travelled 13 nodes