The issue to be brought up with this graph is that Distance and Cost are not appearing on it. This is due to a bug in the code that prevents the program from properly storing the data for these data sets. The issue is being investigated but could not be solved in time for the due date. As it stands, they are being counted as 0 for all data sets.

As for the actual performance of each algorithm, Dijkstra and A\* (or A Star as it has been labeled) routinely performed better than DFS or BFS. They found the path the fastest and found the least number of nodes required between the source and the destination. The trade off is that these search algorithms took a much longer time to search through the data set. Additionally, the Dijkstra performed slightly better overall than the A\*, but this is likely because it was tested on such a small data set. Because the A\* has extra calculations to perform with each search, it slightly raises the amount of time it takes to find the next node. If the graph was larger, then I believe the increased time it takes to find the next node will outpace the Dijkstra search.

In general, the Iterative and Recursive functions of the DFS and BFS algorithms were similar to their respective base algorithm class. Of the two though, BFS outperformed the DFS solution in most cases. One aspect about the BFS Iterative solution is that its normalized Nodes Explored set is much higher than any other section of the graph. This means that the average number of nodes explored is almost always the maximum. This contrasts with its recursive function, which has the lowest normalization number for Nodes Explored. The exact cause of this is unexplained, but there is very likely some difference between the two functions that causes them to act differently.