# Source to Destination

Adjacency List:

| | DFS Iterative | DFS Recursive | BFS Iterative | BFS Recursive | Dijkstra | A* |
|---|---|---|---|---|---|---|
| Nodes in Path | 8.22 | 9.51 | 3.67 | 3.6 | 3.8 | 3.72 |
| Nodes Explored | 8.22 | 9.51 | 8.85 | 9.22 | 9.95 | 9.55 |
| Time (sec) | 0.0000290491 | 0.000024442 | 0.0000270834 | 0.000017651 | 0.0000775063 | 0.0000645461 |
| Distance | 7.22 | 8.51 | 2.67 | 2.6 | 2.8 | 2.72 |
| Cost | 18.66 | 21.85 | 6.75 | 6.44 | 6.93 | 6.82 |

Adjacency Matrix:

| | DFS Iterative | DFS Recursive | BFS Iterative | BFS Recursive | Dijkstra | A* |
|---|---|---|---|---|---|---|
| Nodes in Path | 9.25 | 8.48 | 3.63 | 3.72 | 3.76 | 3.44 |
| Nodes Explored | 9.25 | 8.48 | 8.35 | 9.69 | 9.65 | 8.56 |
| Time (sec) | 0.0000139679 | 0.0000072335 | 0.0000421069 | 0.0000151748 | 0.0000578152 | 0.0000509673 |
| Distance | 8.25 | 7.48 | 2.63 | 2.72 | 2.76 | 2.44 |
| Cost | 20.11 | 18.43 | 6.61 | 6.74 | 6.76 | 6.03 |

# Destination to Source

Adjacency List:

| | DFS Iterative | DFS Recursive | BFS Iterative | BFS Recursive | Dijkstra | A* |
|---|---|---|---|---|---|---|
| Nodes in Path | 9.04 | 9.53 | 3.67 | 3.6 | 3.8 | 3.72 |
| Nodes Explored | 9.04 | 9.53 | 8.51 | 8.79 | 9.9 | 9.67 |
| Time (sec) | 0.0000334015 | 0.0000111964 | 0.0000117292 | 0.0000195812 | 0.0000431373 | 0.0000587834 |
| Distance | 8.04 | 8.53 | 2.67 | 2.6 | 2.8 | 2.72 |
| Cost | 20.66 | 21.73 | 6.75 | 6.44 | 6.93 | 6.82 |

Adjacency Matrix:

| | DFS Iterative | DFS Recursive | BFS Iterative | BFS Recursive | Dijkstra | A* |
|---|---|---|---|---|---|---|
| Nodes in Path | 8.53 | 8.7 | 3.63 | 3.72 | 3.76 | 3.44 |
| Nodes Explored | 8.53 | 8.7 | 8.4 | 9.24 | 9.43 | 8.81 |
| Time (sec) | 0.0000160134 | 0.0000060342 | 0.0000168536 | 0.0000134379 | 0.0000649969 | 0.0000437706 |
| Distance | 7.53 | 7.7 | 2.63 | 2.72 | 2.76 | 2.44 |
| Cost | 18.45 | 18.57 | 6.61 | 6.74 | 6.76 | 6.03 |

For the amount of nodes in the path, DFS iterative and DFS recursive return the most nodes on average, with the average falling between 8 and 10. The other four algorithm's have average path lengths of around 4 nodes. BFS recursive seems to reach its lowest amount of nodes on average when the search is being performed over a matrix. The amount of nodes explored on average by each algorithm is around the same, falling somewhere between 8 to 10 nodes. The algorithm that seems to be the quickest on average is DFS recursive. I think this algorithm would not be as fast in comparison on a larger data set. The algorithms that take are longest are Dijkstra and A*. This is definitely because Dijkstra and A* are smart algorithms, both using a heuristic which takes more time. Although DFS recursive may be the quickest, it also usually returns a high amount of nodes in the path, so it is not the most efficient algorithm. Furthermore, when path distances are compared, DFS iterative and DFS recursive are once again on the less efficient end of the spectrum. The average distance for these two is between 7 and 10, and the average distance for the other four algorithms is around 3. Since this data set was rather small, it's hard to see the true difference between BFS iterative, BFS recursive, A*, and Dijkstra; On the the graphs their results are pretty similar. I think A* or Dijkstra would be the most efficient algorithm on a larger data set, especially when it comes to distance and cost. It's hard to determine which algorithm is the most "efficient" because this word could mean many different things.