|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Adjacency List(Src->Dest)** | **DFS Iter** | **DFS Rec** | **BFS Iter** | **BFS Rec** | **Dijkstra** | **A\*** |
| Nodes in path | 0.373889 | 0.435556 | 0.518 | 0.494 | 0.447692 | 0.466 |
| Nodes Explored | 0.540417 | 0.605 | 0.696667 | 0.681333 | 0.447692 | 0.466 |
| Execution Time | 0.340833 | 0.375714 | 0.1225 | 0.374615 | 0 | 0.21 |
| Distance | 0.353235 | 0.412941 | 0.469667 | 0.465217 | 0.4 | 0.425 |
| Cost | N/A | N/A | N/A | N/A | N/A | 0.304434 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Adjacency List(Dest->src)** | **DFS Iter** | **DFS Rec** | **BFS Iter** | **BFS Rec** | **Dijkstra** | **A\*** |
| Nodes in path | 0.417059 | 0.398333 | 0.506667 | 0.494667 | 0.444615 | 0.466 |
| Nodes Explored | 0.542917 | 0.565833 | 0.680667 | 0.662 | 0.444615 | 0.466 |
| Execution Time | 0.386667 | 0.3825 | 0.348 | 0.421667 | 0 | 0.308 |
| Distance | 0.400313 | 0.377353 | 0.474348 | 0.462174 | 0.4 | 0.43625 |
| Cost | N/A | N/A | N/A | N/A | N/A | 0.313448 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Adjacency Matrix(Src->Dest)** | **DFS Iter** | **DFS Rec** | **BFS Iter** | **BFS Rec** | **Dijkstra** | **A\*** |
| Nodes in path | 0.485 | 0.486 | 0.485333 | 0.546 | 0.299167 | 0.408333 |
| Nodes Explored | 0.586667 | 0.559167 | 0.654667 | 0.708667 | 0.299167 | 0.408333 |
| Execution Time | 0.258333 | 0.435833 | 0.146667 | 0.226667 | 0 | 0.200833 |
| Distance | 0.481667 | 0.452917 | 0.398148 | 0.474 | 0.4625 | 0.411111 |
| Cost | N/A | N/A | N/A | N/A | N/A | 0.120837 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Adjacency Matrix(Dest->src)** | **DFS Iter** | **DFS Rec** | **BFS Iter** | **BFS Rec** | **Dijkstra** | **A\*** |
| Nodes in path | 0.491333 | 0.431667 | 0.469333 | 0.506667 | 0.3075 | 0.408333 |
| Nodes Explored | 0.559583 | 0.5775 | 0.634667 | 0.669333 | 0.3075 | 0.408333 |
| Execution Time | 0.305836 | 0.36 | 0.12 | 0.1825 | 0 | 0.351429 |
| Distance | 0.45875 | 0.451154 | 0.382593 | 0.425185 | 0.4625 | 0.411111 |
| Cost | N/A | N/A | N/A | N/A | N/A | 0.168859 |

Report:

While there was some difference in performance between the Breadth First Search and Depth First Search, it was not significant. This was most likely since the width and depth of the graph was the same, so neither one had an advantage. But if the graph was deeper I would most likely see more success with the BFS search, and if it was wider, the DFS would probably be better.

The recursive version of both the DFS and BFS, at best did the same as their Iterative Counterparts, while at worse, taking longer in all categories. However, I mainly attribute this to how I implemented them rather, than a recursive implementation being worse.

The A\* search consistently performed better than Dijkstra, which is not surprising considering that it implemented the Heuristic function, which calculated the cost of traversal and helped it find the shortest path. They both did better than the other four algorithms as they attempted to find the shortest path, rather than just iterate through the graph by depth or width.