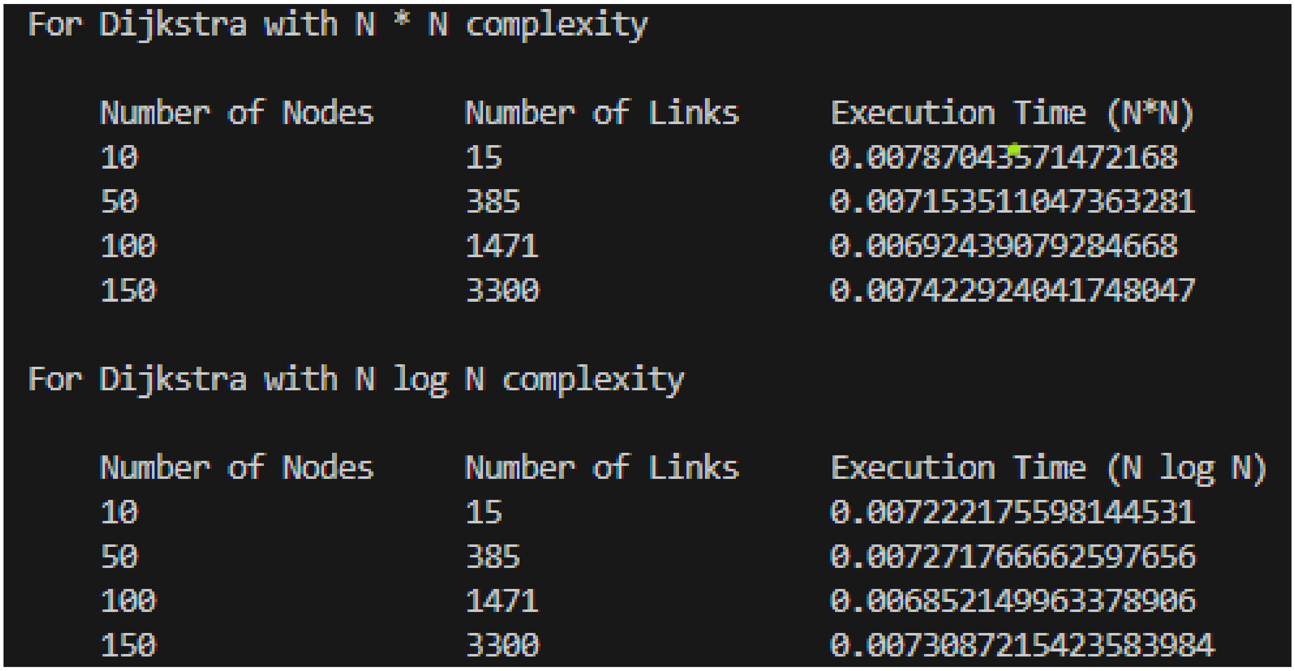
Dijkstra’s Algorithm Complexity Analysis

This report analyses the runtime behaviors of two implementations of Dijkstra's algorithm, with time complexity $O(N^2)$ and the other $O(NlogN)$, verifying if they adhere to their respective complexity bounds.

Method

Both Python implementations of the algorithm were tested on networks of varying sizes using an automated script, run by entering ‘python run_script.py’ in the terminal. Networks were randomly generated with different numbers of nodes and links, ranging from 10 to 150 nodes. The probability of links between each node in the network can be adjusted in the ‘generate_network()’ function and was set to 0.3 to produce these results.

Result



```
For Dijkstra with N * N complexity

    Number of Nodes     Number of Links     Execution Time (N*N)
    10                  15                  0.007870435571472168
    50                  385                 0.007153511047363281
    100                 1471                0.006924390079284668
    150                 3300                0.007422924041748047

For Dijkstra with N log N complexity

    Number of Nodes     Number of Links     Execution Time (N log N)
    10                  15                  0.007222175598144531
    50                  385                 0.007271766662597656
    100                 1471                0.006852149963378906
    150                 3300                0.0073087215423583984
```

The table shows a summary of the run-time information printed by the script where it was noted that as the density of the network increased, the implementation of Dijkstra’s algorithm with a time complexity $O(NlogN)$ exhibited slightly faster execution times.