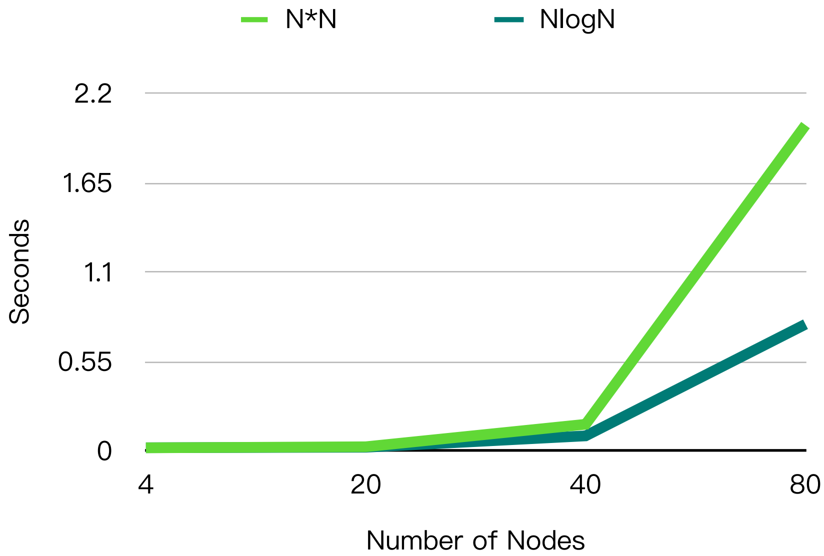
Graph:

```
For Dijkstra with N * N complexity
Number of Nodes      Number of Links      Execution Time (N*N)
4                    4                     0.016733169555664062 seconds
20                   23                    0.023398160934448242 seconds
40                   44                    0.1616871356964113 seconds
80                   84                    2.002807140350342 seconds
For Dijkstra with N log N complexity
Number of Nodes      Number of Links      Execution Time (N log N)
4                    4                     0.01745295524597168 seconds
20                   23                    0.020431995391845703 seconds
40                   44                    0.09028196334838867 seconds
80                   84                    0.7772500514984131 seconds
```



The graph above illustrates my two Dijkstra methods. I used a total of 4 test cases to test:

Test Case 1: 4 nodes, 4 links
Test Case 2: 20 nodes, 23 links
Test Case 3: 40 nodes, 44 links
Test Case 4: 80 nodes, 84 links

Complexity analyse:

- With small graphs (e.g., 4 nodes), the difference in execution time is negligible. However, as the graph size increases, the N log N implementation significantly outperforms the N*N implementation. For example, at 80 nodes, the execution time for the N*N implementation is approximately 2 seconds, while the N log N implementation is less than 0.8 seconds.
- The N*N implementation shows a steep increase in execution time as the number of nodes increases. This is expected because the complexity grows quadratically.
- The N log N implementation has a much slower increase in execution time, demonstrating the logarithmic component's efficiency, especially evident as the node count scales up.
- The test results validate the theoretical time complexity. For larger graphs, the N log N implementation is clearly more efficient, making it more suitable for applications involving large networks.

Usage: (Type in terminal)

Dijkstra: `./Dijkstra < test_case.txt` (You can modify the content in test_case.txt)
DijkstraNlogN: `./DijkstraNlogN < test_case.txt`
test_dijkstra_performance: `python3 test_dijkstra_performance.py`