PROJECT : Dijkstra Implementation

Details: Applying Dijkstra without priority queue and by my own logic

Implementation: I implement this algorithm n cpp and py for efficiency checking and visualize the graph wit reaTitle: Implementation of Dijkstra's Algorithm

Objective: Implementing Dijkstra's algorithm without using a priority queue, utilizing a custom logic to find the shortest path in weighted graphs.

Scope:

- Algorithm implemented in C++ and Python to analyze efficiency and performance differences.
- Graph visualization created using React.js and JavaScript.

Implementation Details

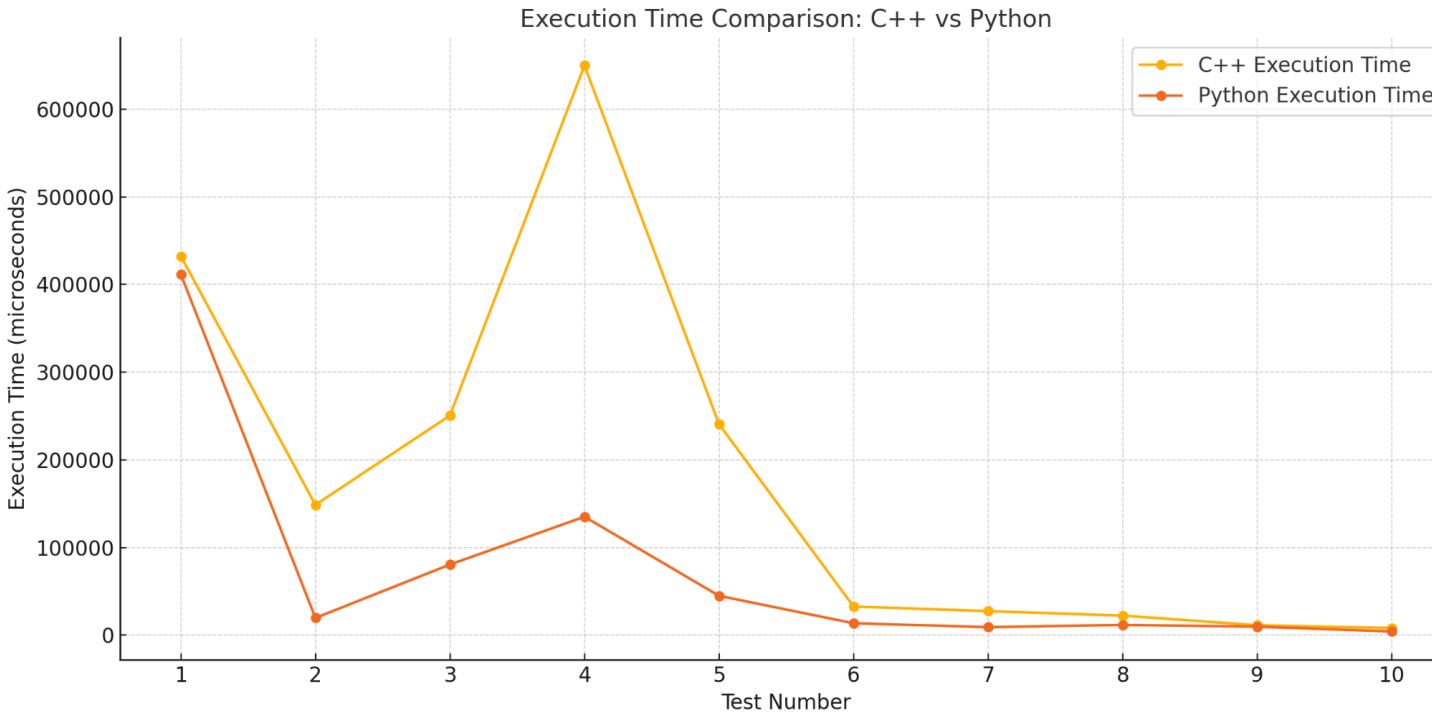
1. Language Comparison:
    - C++ and Python were selected to compare execution times for various test cases.
    - Each implementation adheres to a brute-force approach, focusing on simplicity and correctness.
2. Visualization:
    - Graphs were visualized using React.js for better understanding of shortest paths and graph connectivity.

Test Cases

- A total of 10 test cases were designed to evaluate the performance of the algorithm.
- Each test case varies in graph size and structure to test execution times and validate correctness.

Group Members

1. Daniyal Fahim (Group Leader)
   Registration Number: 22K-4282
2. Daiyan Ur Rehman
   Registration Number: 22K-4167
3. Dawood Adnan
   Registration Number: 22K-4663



Graph Analysis

The graphs generated from the test data illustrate:

1. C++ spikes in execution time for larger graphs due to adjacency list inefficiencies.
2. Python's smoother execution trend, though slower for smaller inputs.

By integrating advanced data structures (e.g., priority queues), both implementations can achieve O((V+E)logV)O((V + E) \log V)O((V+E)logV), eliminating observed anomalies and improving scalability.

1. Execution Time Trends:

- C++ generally has faster execution times than Python across most tests.
- However, in certain larger cases, C++ shows unexpectedly high execution times compared to Python.

2. Test Results Summary:

- Small Graphs: C++ consistently outperforms Python due to its highly optimized compiler and low-level memory access.

    - Larger Graphs: In some tests, C++ shows performance spikes compared to Python.

1. Compiler and Runtime Differences

- C++: Compiled to machine code, optimizing performance for numerical and iterative operations.
- Python: Interpreted language, inherently slower due to dynamic typing and garbage collection overhead.

2. Algorithm Complexity

- Both implementations have O(V2)O(V^2)O(V2) complexity due to the brute-force selection of the next node.
- Python's implementation may perform better on certain test cases due to differences in how adjacency lists and loops are processed in memory.

3. Memory Management

- C++: Directly manages memory, but improper handling of data structures (e.g., adjacency lists) can cause inefficiencies for larger data.
- Python: Uses automatic memory management (garbage collection), which may reduce overhead in small test cases.

4. Input Handling

- Python's dictionaries and lists are optimized for sparse data, potentially handling adjacency lists more efficiently than raw arrays or linked lists in C++.

5. Timing Anomalies

- C++ Timing Hike: Likely caused by how the `std::chrono` library captures time intervals, or due to overhead from dynamic memory allocation in `std::list` operations.
- Python: `time.perf_counter` is less prone to such anomalies.

1. **C++ Strengths:**
    - Optimal for compute-intensive tasks.
    - Shows consistent superiority for smaller graphs.
2. **Python Strengths:**

    - Better handling of sparse adjacency lists in certain scenarios.
    - Lower memory management overhead can outperform C++ for specific graph configurations.

Test Results

Test #1

- **CPP**: 432,254 µs
- **Python**: 411,553.6 µs
- **Python is ~4.78% faster than CPP**

Test #2

- **CPP**: 148,696 µs
- **Python**: 19,764.7 µs
- **Python is ~87.71% faster than CPP**

Test #3

- **CPP**: 250,612 µs
- **Python**: 80,676.1 µs
- **Python is ~67.79% faster than CPP**

Test #4

- **CPP**: 649,689 µs
- **Python**: 135,166.7 µs
- **Python is ~79.20% faster than CPP**

Test #5

- **CPP**: 240,703 µs
- **Python**: 44,986.2 µs
- **Python is ~81.30% faster than CPP**

Test #6

- **CPP**: 32,783 µs
- **Python**: 13,680.2 µs
- **Python is ~58.26% faster than CPP**

Test #7

- **CPP**: 27,506 µs
- **Python**: 9,298.6 µs
- **Python is ~66.20% faster than CPP**

Test #8

- **CPP**: 22,427 µs
- **Python**: 11,755.6 µs
- **Python is ~47.59% faster than CPP**

Test #8 b

- **CPP**: 11,406 µs
- **Python**: 9,848.9 µs
- **Python is ~13.64% faster than CPP**

Test #9

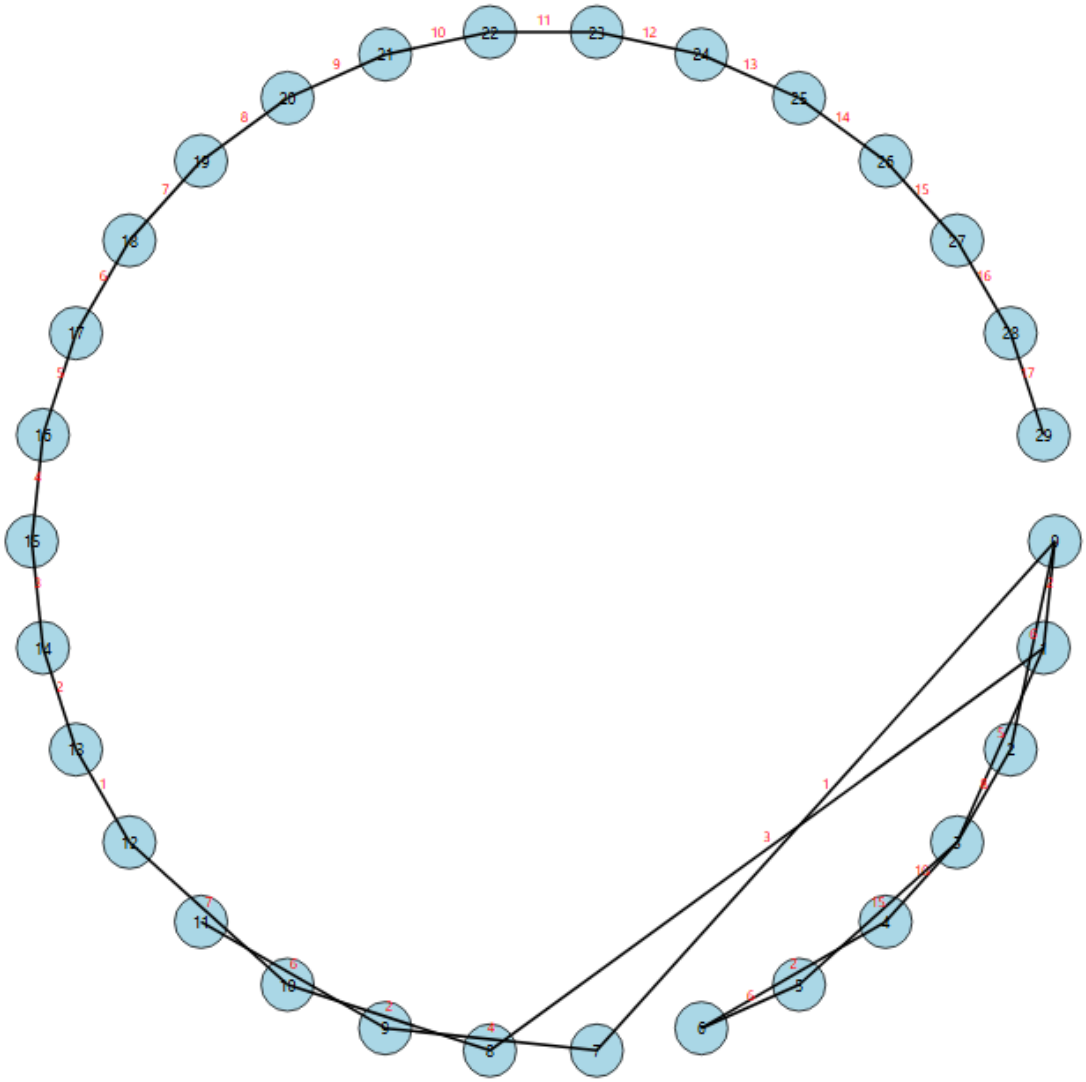
- **CPP**: 190,365 µs
- **Python**: 38,823.1 µs
- **Python is ~79.61% faster than CPP**

Test #10

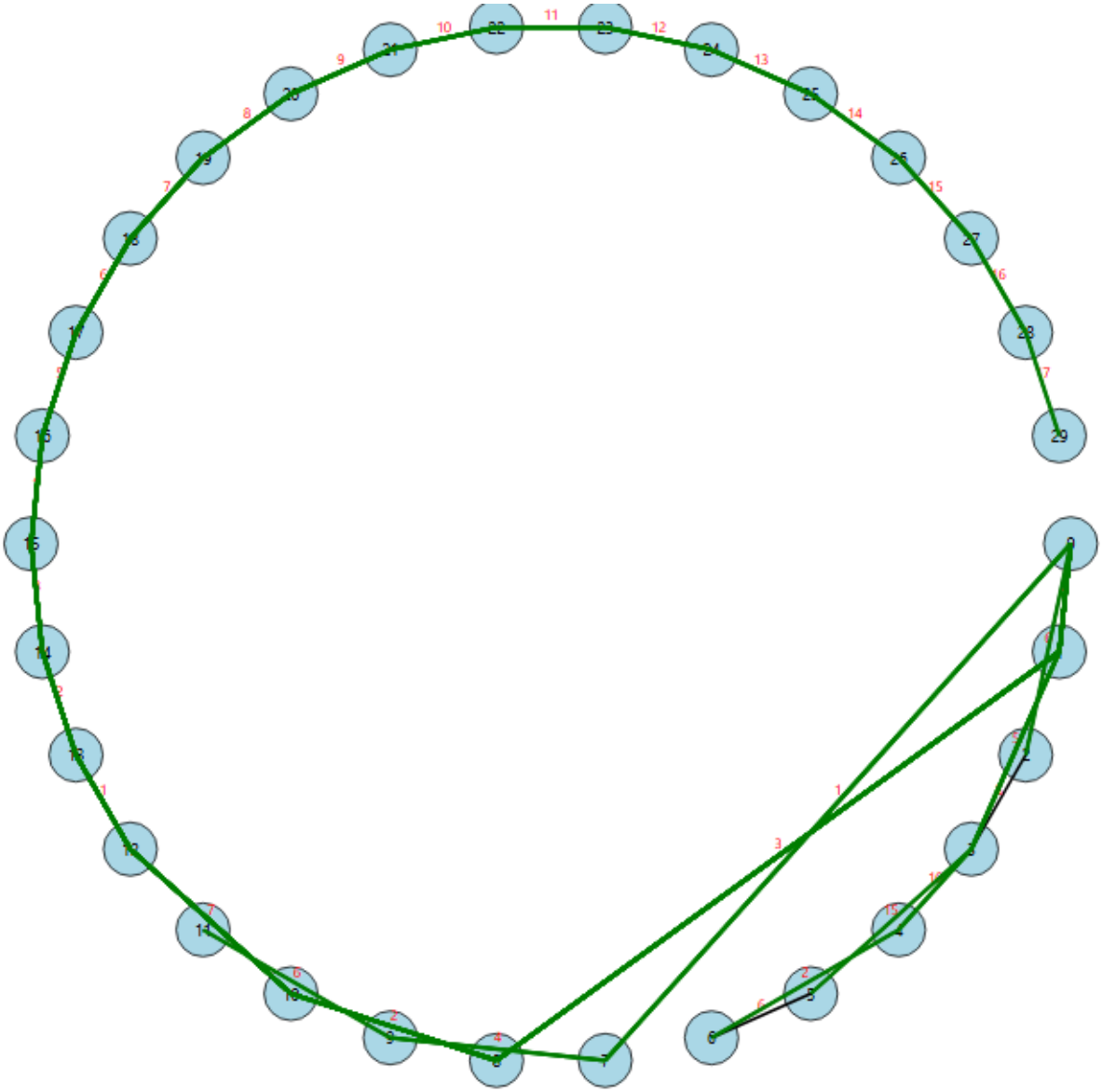
- **CPP**: 8,209 µs
- **Python**: 4,287.2 µs
- **Python is ~47.77% faster than CPP**

TEST # 1

```
addEdge(0, 1, 2);
addEdge(0, 2, 6);
addEdge(1, 3, 5);
addEdge(2, 3, 8);addEdge(3, 4, 10);
addEdge(3, 5, 15);addEdge(4, 6, 2);
addEdge(5, 6, 6);addEdge(0, 7, 1);
addEdge(1, 8, 3);addEdge(7, 9, 4);
addEdge(8, 10, 2);addEdge(9, 11, 6);
addEdge(10, 12, 7);addEdge(12, 13, 1);
addEdge(13, 14, 2);addEdge(14, 15, 3);
addEdge(15, 16, 4);addEdge(16, 17, 5);
addEdge(17, 18, 6);addEdge(18, 19, 7);
addEdge(19, 20, 8);addEdge(20, 21, 9);
addEdge(21, 22, 10);addEdge(22, 23, 11);
addEdge(23, 24, 12);addEdge(24, 25, 13);
addEdge(25, 26, 14);addEdge(26, 27, 15);
addEdge(27, 28, 16); addEdge(28, 29, 17);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|1|3|4 (Distance: 3)
- Path to 5: 0|1|3|5 (Distance: 3)
- Path to 6: 0|1|3|4|6 (Distance: 4)
- Path to 7: 0|7 (Distance: 1)
- Path to 8: 0|1|8 (Distance: 2)
- Path to 9: 0|7|9 (Distance: 2)
- Path to 10: 0|1|8|10 (Distance: 3)
- Path to 11: 0|7|9|11 (Distance: 3)
- Path to 12: 0|1|8|10|12 (Distance: 4)
- Path to 13: 0|1|8|10|12|13 (Distance: 5)
- Path to 14: 0|1|8|10|12|13|14 (Distance: 6)
- Path to 15: 0|1|8|10|12|13|14|15 (Distance: 7)
- Path to 16: 0|1|8|10|12|13|14|15|16 (Distance: 8)
- Path to 17: 0|1|8|10|12|13|14|15|16|17 (Distance: 9)
- Path to 18: 0|1|8|10|12|13|14|15|16|17|18 (Distance: 10)
- Path to 19: 0|1|8|10|12|13|14|15|16|17|18|19 (Distance: 11)
- Path to 20: 0|1|8|10|12|13|14|15|16|17|18|19|20 (Distance: 12)
- Path to 21: 0|1|8|10|12|13|14|15|16|17|18|19|20|21 (Distance: 13)
- Path to 22: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22 (Distance: 14)
- Path to 23: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23 (Distance: 15)
- Path to 24: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24 (Distance: 16)
- Path to 25: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24|25 (Distance: 17)
- Path to 26: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26 (Distance: 18)
- Path to 27: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27 (Distance: 19)
- Path to 28: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28 (Distance: 20)
- Path to 29: 0|1|8|10|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29 (Distance: 21)

CPP (Execution time: 432254 microseconds)

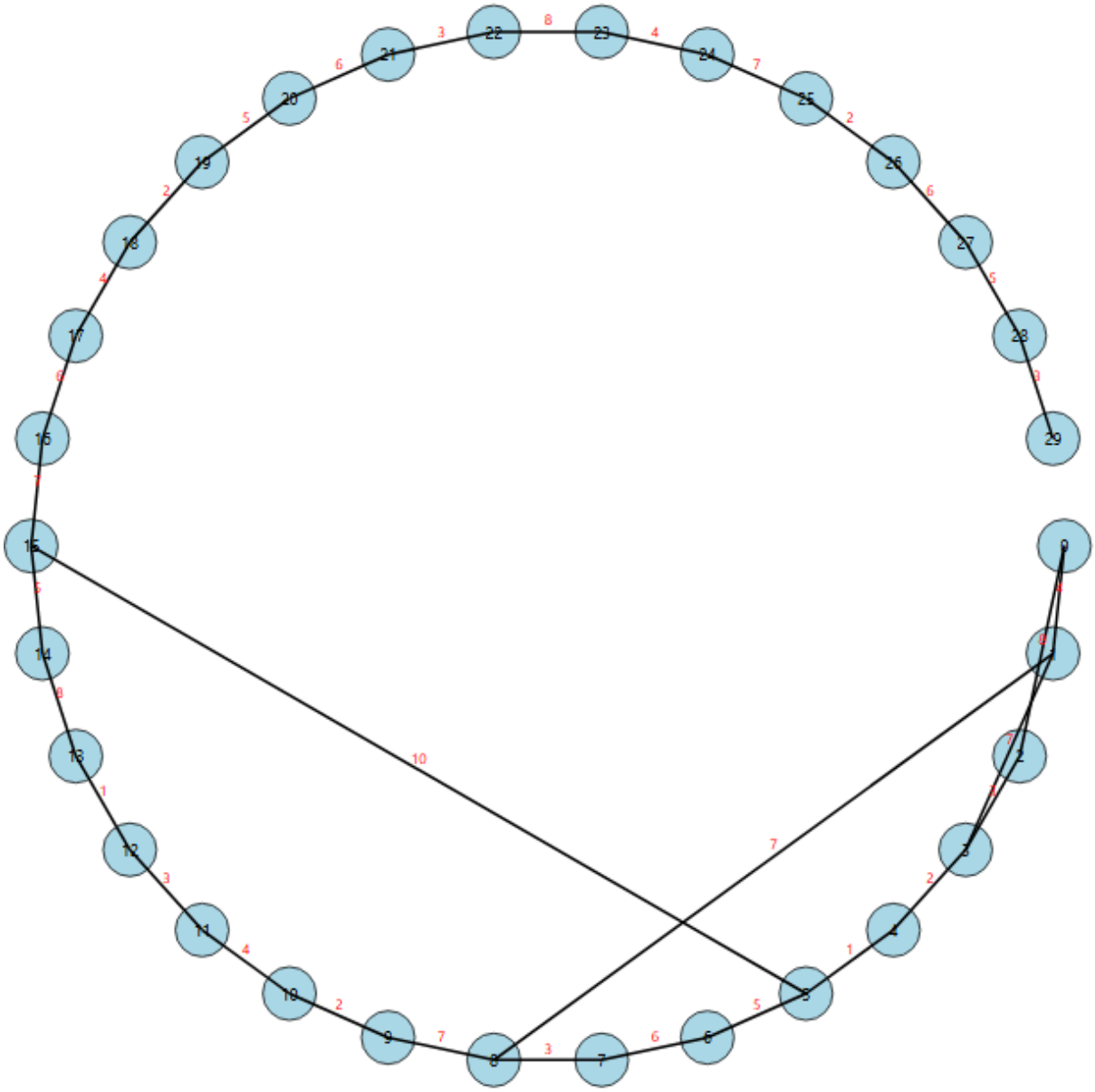
```
Vertex   Distance       Path
0               0               0
1               2               0 -> 1
2               6               0 -> 2
3               7               0 -> 1 -> 3
4               17              0 -> 1 -> 3 -> 4
5               22              0 -> 1 -> 3 -> 5
6               19              0 -> 1 -> 3 -> 4 -> 6
7               1               0 -> 7
8               5               0 -> 1 -> 8
9               5               0 -> 7 -> 9
10              7               0 -> 1 -> 8 -> 10
11              11              0 -> 7 -> 9 -> 11
12              14              0 -> 1 -> 8 -> 10 -> 12
13              15              0 -> 1 -> 8 -> 10 -> 12 -> 13
14              17              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14
15              20              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15
16              24              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16
17              29              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17
18              35              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18
19              42              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19
20              50              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              59              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
22              69              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
23              80              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
24              92              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
25              105             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25
26              119             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26
27              134             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27
28              150             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28

29              167             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28
 -> 29
```

PYTHON (Execution time: 411553.600 microseconds)

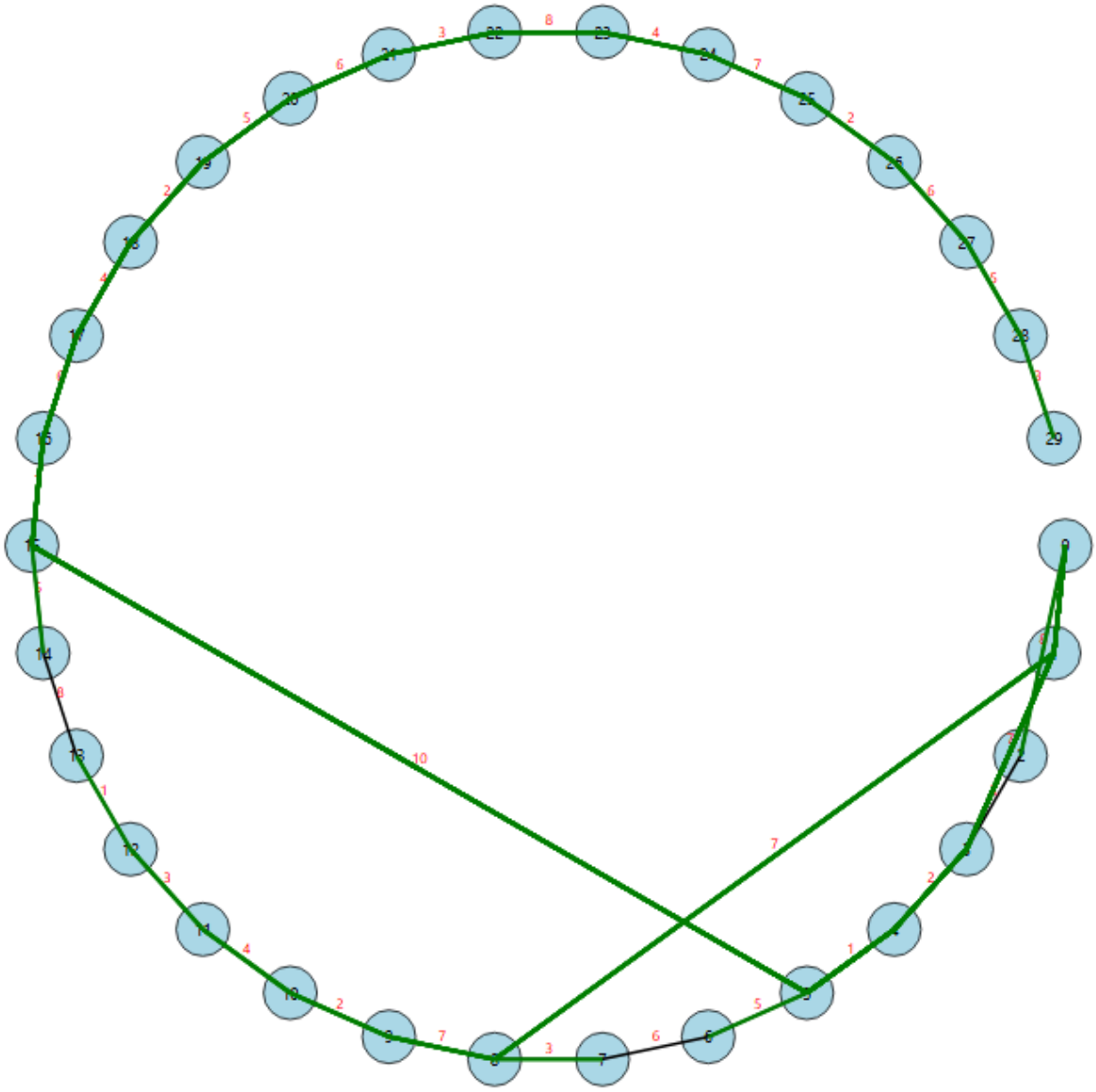
```
 Vertex  Distance        Path
 0               0               0
 1               2               0 -> 1
 2               6               0 -> 2
 3               7               0 -> 1 -> 3
 4               17              0 -> 1 -> 3 -> 4
 5               22              0 -> 1 -> 3 -> 5
 6               19              0 -> 1 -> 3 -> 4 -> 6
 7               1               0 -> 7
 8               5               0 -> 1 -> 8
 9               5               0 -> 7 -> 9
 10              7               0 -> 1 -> 8 -> 10
 11              11              0 -> 7 -> 9 -> 11
 12              14              0 -> 1 -> 8 -> 10 -> 12
 13              15              0 -> 1 -> 8 -> 10 -> 12 -> 13
 14              17              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14
 15              20              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15
 16              24              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16
 17              29              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17
 18              35              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18
 19              42              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19
 20              50              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
 21              59              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
 22              69              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
 23              80              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
 24              92              0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
 25              105             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -
> 25
 26              119             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -
> 25 -> 26
 27              134             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27
 28              150             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28

 29              167             0 -> 1 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28
  -> 29
 Execution time: 411553.600 microseconds
```

TEST # 2

```
    // Adding edges with weights
g.addEdge(0, 1, 4); g.addEdge(0, 2, 8); g.addEdge(1, 3, 7); g.addEdge(2, 3, 3);
g.addEdge(3, 4, 2); g.addEdge(4, 5, 1); g.addEdge(5, 6, 5); g.addEdge(6, 7, 6);
g.addEdge(7, 8, 3); g.addEdge(8, 9, 7); g.addEdge(9, 10, 2); g.addEdge(10, 11, 4);
g.addEdge(11, 12, 3); g.addEdge(12, 13, 1); g.addEdge(13, 14, 8); g.addEdge(14, 15, 5);
g.addEdge(15, 16, 7); g.addEdge(16, 17, 6); g.addEdge(17, 18, 4); g.addEdge(18, 19, 2);
g.addEdge(19, 20, 5); g.addEdge(20, 21, 6); g.addEdge(21, 22, 3); g.addEdge(22, 23, 8);
g.addEdge(23, 24, 4); g.addEdge(24, 25, 7); g.addEdge(25, 26, 2); g.addEdge(26, 27, 6);
g.addEdge(27, 28, 5); g.addEdge(28, 29, 3); g.addEdge(1, 8, 7); g.addEdge(5, 15, 10);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|1|3|4 (Distance: 3)
- Path to 5: 0|1|3|4|5 (Distance: 4)
- Path to 6: 0|1|3|4|5|6 (Distance: 5)
- Path to 7: 0|1|8|7 (Distance: 3)
- Path to 8: 0|1|8 (Distance: 2)
- Path to 9: 0|1|8|9 (Distance: 3)
- Path to 10: 0|1|8|9|10 (Distance: 4)
- Path to 11: 0|1|8|9|10|11 (Distance: 5)
- Path to 12: 0|1|8|9|10|11|12 (Distance: 6)
- Path to 13: 0|1|8|9|10|11|12|13 (Distance: 7)
- Path to 14: 0|1|3|4|5|15|14 (Distance: 6)
- Path to 15: 0|1|3|4|5|15 (Distance: 5)
- Path to 16: 0|1|3|4|5|15|16 (Distance: 6)
- Path to 17: 0|1|3|4|5|15|16|17 (Distance: 7)
- Path to 18: 0|1|3|4|5|15|16|17|18 (Distance: 8)
- Path to 19: 0|1|3|4|5|15|16|17|18|19 (Distance: 9)
- Path to 20: 0|1|3|4|5|15|16|17|18|19|20 (Distance: 10)
- Path to 21: 0|1|3|4|5|15|16|17|18|19|20|21 (Distance: 11)
- Path to 22: 0|1|3|4|5|15|16|17|18|19|20|21|22 (Distance: 12)
- Path to 23: 0|1|3|4|5|15|16|17|18|19|20|21|22|23 (Distance: 13)
- Path to 24: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24 (Distance: 14)
- Path to 25: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24|25 (Distance: 15)
- Path to 26: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24|25|26 (Distance: 16)
- Path to 27: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24|25|26|27 (Distance: 17)
- Path to 28: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24|25|26|27|28 (Distance: 18)
- Path to 29: 0|1|3|4|5|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29 (Distance: 19)

CPP (Execution time: 148696 microseconds)

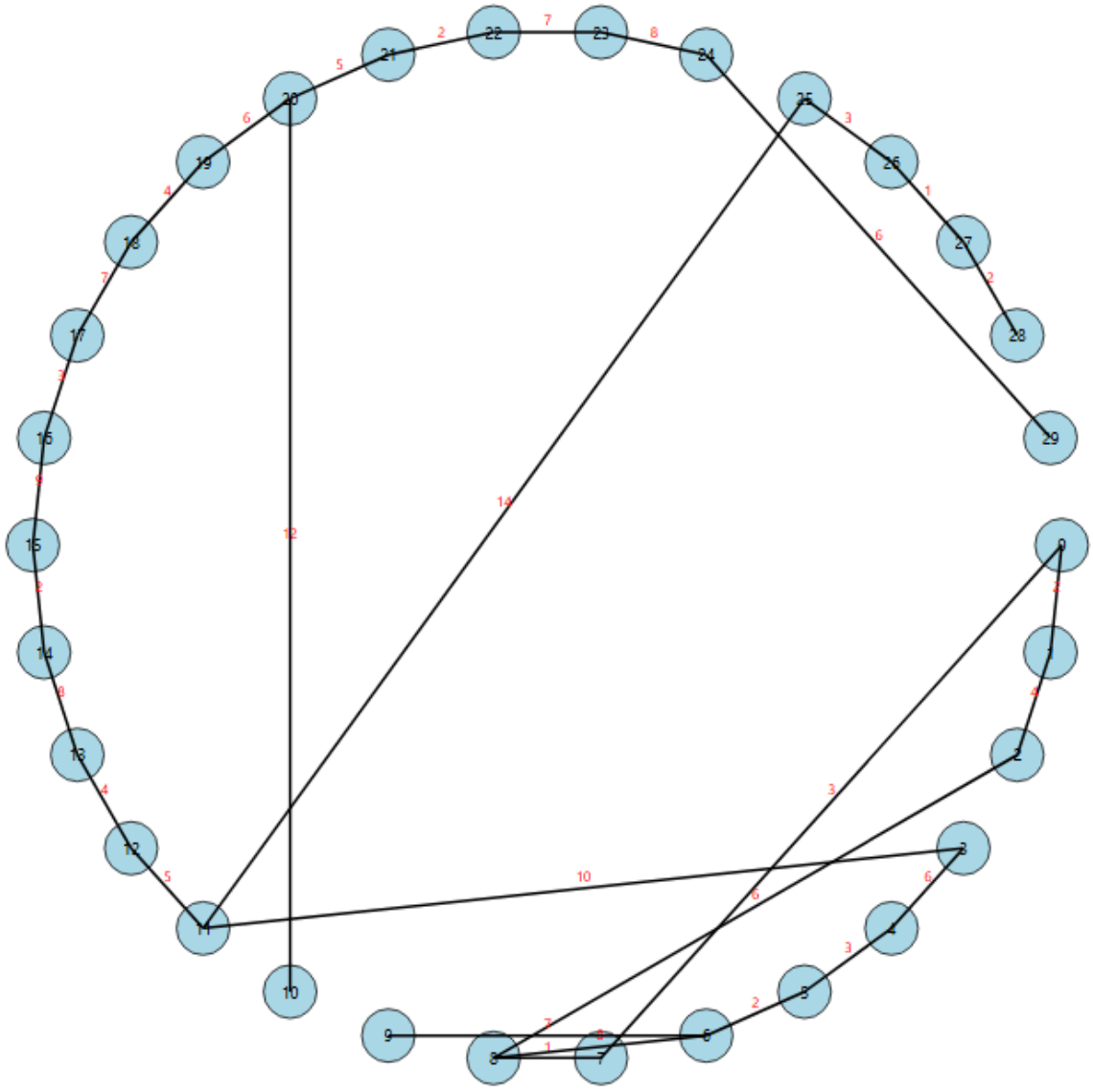
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> cd "e:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src\" ; if ($?) { g++ dij
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0               0               0
1               4               0 -> 1
2               8               0 -> 2
3               11              0 -> 1 -> 3
4               13              0 -> 1 -> 3 -> 4
5               14              0 -> 1 -> 3 -> 4 -> 5
6               19              0 -> 1 -> 3 -> 4 -> 5 -> 6
7               14              0 -> 1 -> 8 -> 7
8               11              0 -> 1 -> 8
9               18              0 -> 1 -> 8 -> 9
10              20              0 -> 1 -> 8 -> 9 -> 10
11              24              0 -> 1 -> 8 -> 9 -> 10 -> 11
12              27              0 -> 1 -> 8 -> 9 -> 10 -> 11 -> 12
13              28              0 -> 1 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13
14              29              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 14
15              24              0 -> 1 -> 3 -> 4 -> 5 -> 15
16              31              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16
17              37              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17
18              41              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18
19              43              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19
20              48              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              54              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
22              57              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
23              65              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
24              69              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
25              76              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25
26              78              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26
27              84              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27
28              89              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28
29              92              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28 -> 29
Execution time: 148696 microseconds
```

PYTHON (Execution time: 19764.700 microseconds)

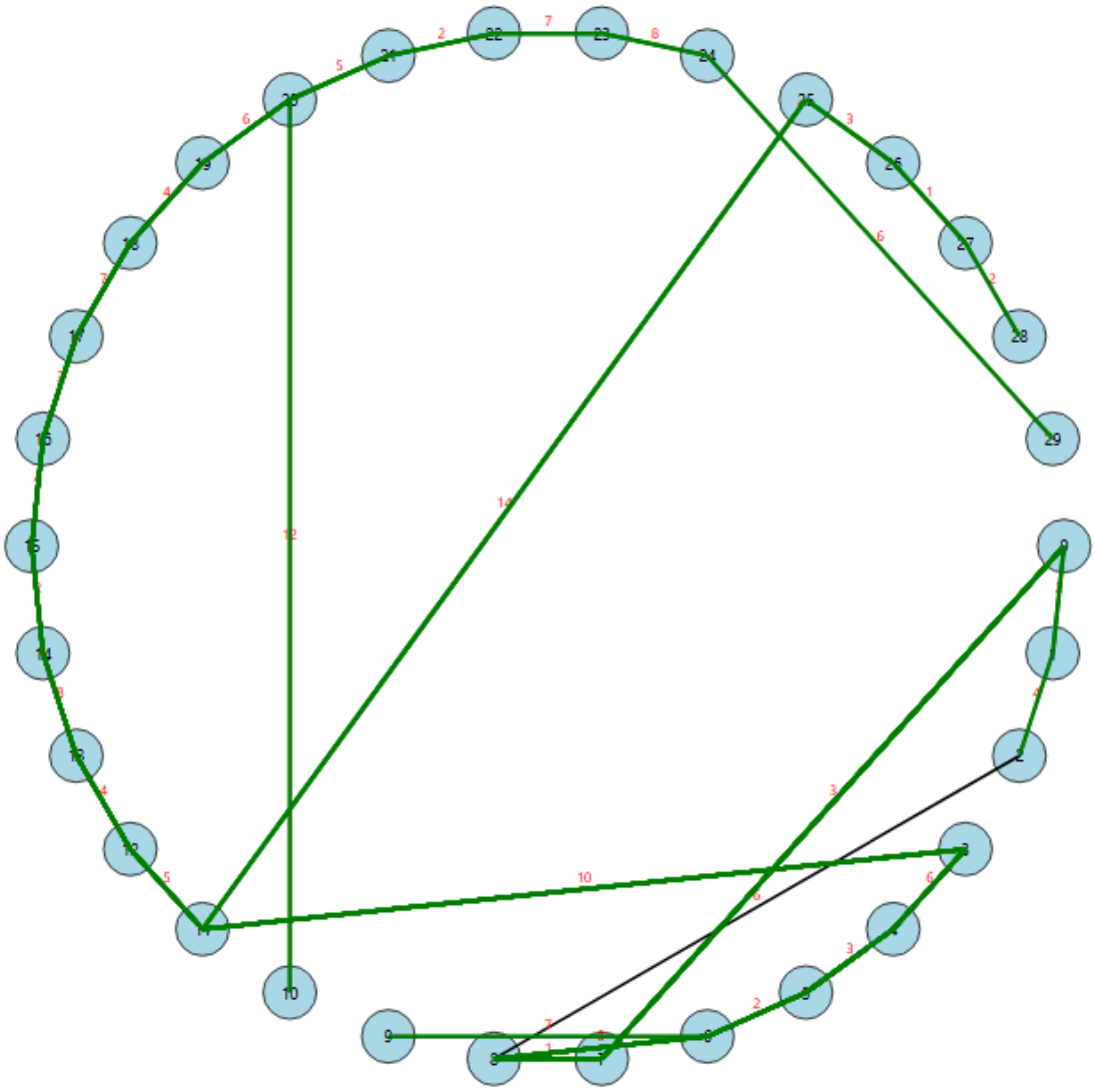
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> python -u "e:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src\dijkstra_python_
implementation.py"
Vertex   Distance       Path
0               0               0
1               4               0 -> 1
2               8               0 -> 2
3               11              0 -> 1 -> 3
4               13              0 -> 1 -> 3 -> 4
5               14              0 -> 1 -> 3 -> 4 -> 5
6               19              0 -> 1 -> 3 -> 4 -> 5 -> 6
7               14              0 -> 1 -> 8 -> 7
8               11              0 -> 1 -> 8
9               18              0 -> 1 -> 8 -> 9
10              20              0 -> 1 -> 8 -> 9 -> 10
11              24              0 -> 1 -> 8 -> 9 -> 10 -> 11
12              27              0 -> 1 -> 8 -> 9 -> 10 -> 11 -> 12
13              28              0 -> 1 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13
14              29              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 14
15              24              0 -> 1 -> 3 -> 4 -> 5 -> 15
16              31              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16
17              37              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17
18              41              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18
19              43              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19
20              48              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              54              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
22              57              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
23              65              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
24              69              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
25              76              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25
26              78              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26
27              84              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27
28              89              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28
29              92              0 -> 1 -> 3 -> 4 -> 5 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 -> 25 -> 26 -> 27 -> 28 -> 29
Execution time: 19764.700 microseconds
```

TEST # 3

```
if (vertices >= 30) {
  addEdge(0, 1, 2); addEdge(0, 7, 3); addEdge(1, 2, 4); addEdge(2, 8, 6);
  addEdge(7, 8, 1); addEdge(8, 6, 8); addEdge(6, 5, 2); addEdge(6, 9, 7);
  addEdge(5, 4, 3); addEdge(4, 3, 6); addEdge(3, 11, 10); addEdge(11, 12, 5);
  addEdge(12, 13, 4); addEdge(13, 14, 8); addEdge(14, 15, 2); addEdge(15, 16, 9);
  addEdge(16, 17, 3); addEdge(17, 18, 7); addEdge(18, 19, 4); addEdge(19, 20, 6);
  addEdge(20, 21, 5); addEdge(21, 22, 2); addEdge(22, 23, 7); addEdge(23, 24, 8);
  addEdge(24, 29, 6); addEdge(25, 26, 3); addEdge(26, 27, 1); addEdge(27, 28, 2);
  addEdge(10, 20, 12); addEdge(11, 25, 14);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|1|2 (Distance: 2)
- Path to 3: 0|7|8|6|5|4|3 (Distance: 6)
- Path to 4: 0|7|8|6|5|4 (Distance: 5)
- Path to 5: 0|7|8|6|5 (Distance: 4)
- Path to 6: 0|7|8|6 (Distance: 3)
- Path to 7: 0|7 (Distance: 1)
- Path to 8: 0|7|8 (Distance: 2)
- Path to 9: 0|7|8|6|9 (Distance: 4)
- Path to 10: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|10 (Distance: 17)
- Path to 11: 0|7|8|6|5|4|3|11 (Distance: 7)
- Path to 12: 0|7|8|6|5|4|3|11|12 (Distance: 8)
- Path to 13: 0|7|8|6|5|4|3|11|12|13 (Distance: 9)
- Path to 14: 0|7|8|6|5|4|3|11|12|13|14 (Distance: 10)
- Path to 15: 0|7|8|6|5|4|3|11|12|13|14|15 (Distance: 11)
- Path to 16: 0|7|8|6|5|4|3|11|12|13|14|15|16 (Distance: 12)
- Path to 17: 0|7|8|6|5|4|3|11|12|13|14|15|16|17 (Distance: 13)
- Path to 18: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18 (Distance: 14)
- Path to 19: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19 (Distance: 15)
- Path to 20: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20 (Distance: 16)
- Path to 21: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|21 (Distance: 17)
- Path to 22: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|21|22 (Distance: 18)
- Path to 23: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|21|22|23 (Distance: 19)
- Path to 24: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|21|22|23|24 (Distance: 20)
- Path to 25: 0|7|8|6|5|4|3|11|25 (Distance: 8)
- Path to 26: 0|7|8|6|5|4|3|11|25|26 (Distance: 9)
- Path to 27: 0|7|8|6|5|4|3|11|25|26|27 (Distance: 10)
- Path to 28: 0|7|8|6|5|4|3|11|25|26|27|28 (Distance: 11)
- Path to 29: 0|7|8|6|5|4|3|11|12|13|14|15|16|17|18|19|20|21|22|23|24|29 (Distance: 21)

CPP (Execution time: 250612 microseconds)

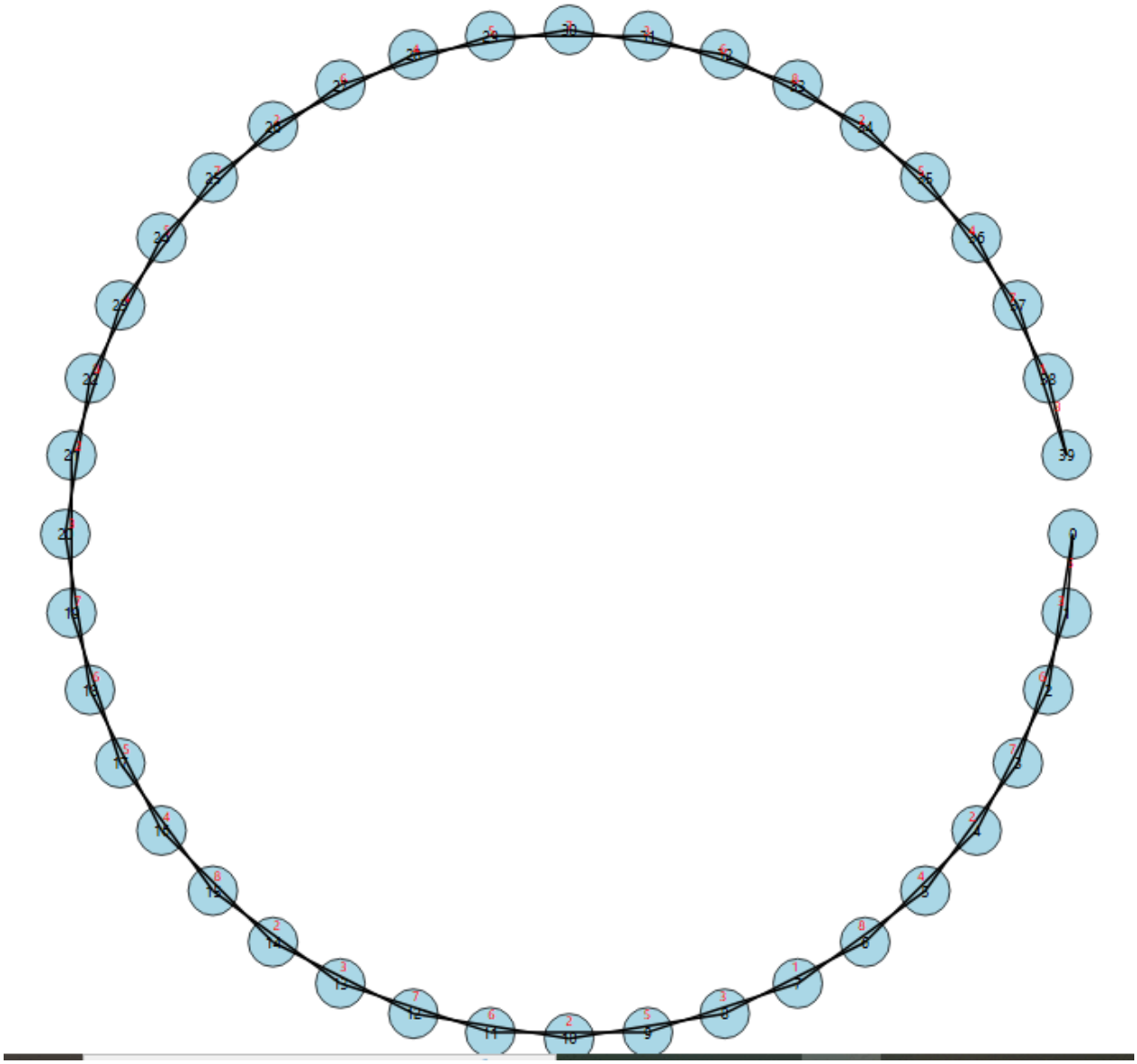
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> cd "e:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src\" ; if ($?) { g++ dijks
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0               0               0
1               2               0 -> 1
2               6               0 -> 1 -> 2
3               23              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3
4               17              0 -> 7 -> 8 -> 6 -> 5 -> 4
5               14              0 -> 7 -> 8 -> 6 -> 5
6               12              0 -> 7 -> 8 -> 6
7               3               0 -> 7
8               4               0 -> 7 -> 8
9               19              0 -> 7 -> 8 -> 6 -> 9
10              93              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 10
11              33              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11
12              38              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12
13              42              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13
14              50              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14
15              52              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15
16              61              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16
17              64              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17
18              71              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18
19              75              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19
20              81              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              86              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
22              88              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
23              95              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
24              103             0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
25              47              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25
26              50              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26
27              51              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26 -> 27
28              53              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26 -> 27 -> 28
29              109             0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 ->
29

Execution time: 250612 microseconds
```

PYTHON (Execution time: 80676.100 microseconds)

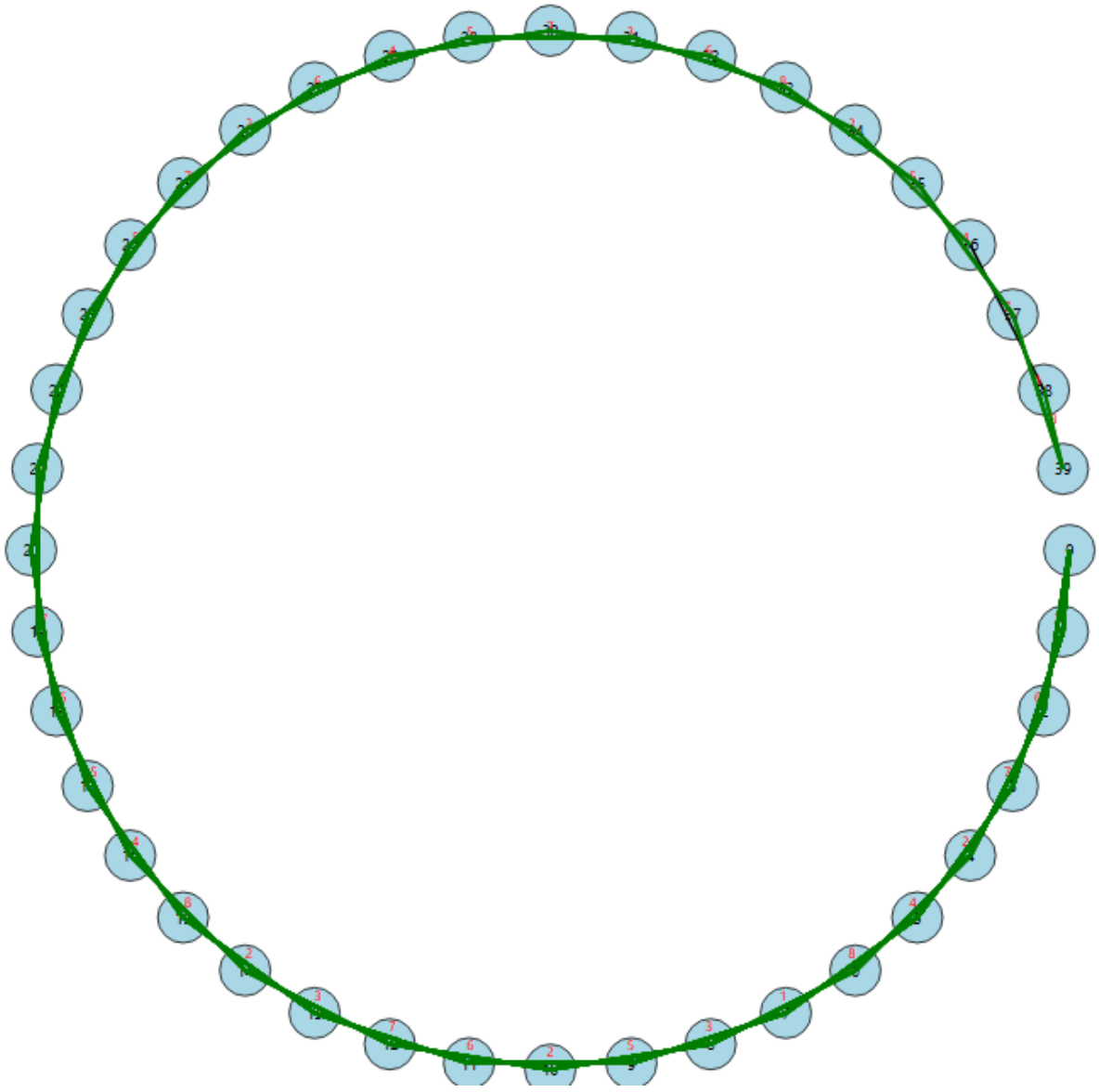
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> python -u "e:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src\dijkstra_python_
implementation.py"
Vertex   Distance       Path
0               0               0
1               2               0 -> 1
2               6               0 -> 1 -> 2
3               23              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3
4               17              0 -> 7 -> 8 -> 6 -> 5 -> 4
5               14              0 -> 7 -> 8 -> 6 -> 5
6               12              0 -> 7 -> 8 -> 6
7               3               0 -> 7
8               4               0 -> 7 -> 8
9               19              0 -> 7 -> 8 -> 6 -> 9
10              93              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 10
11              33              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11
12              38              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12
13              42              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13
14              50              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14
15              52              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15
16              61              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16
17              64              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17
18              71              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18
19              75              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19
20              81              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              86              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21
22              88              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22
23              95              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23
24              103             0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24
25              47              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25
26              50              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26
27              51              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26 -> 27
28              53              0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 25 -> 26 -> 27 -> 28
29              109             0 -> 7 -> 8 -> 6 -> 5 -> 4 -> 3 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 21 -> 22 -> 23 -> 24 ->
29

Execution time: 80676.100 microseconds
```

TEST # 4

```
        addEdge(0, 1, 5);
 addEdge(0, 2, 3);
 addEdge(1, 3, 6);
 addEdge(2, 4, 7);
 addEdge(3, 5, 2);
 addEdge(4, 6, 4);
 addEdge(5, 7, 8);
 addEdge(6, 8, 1);
 addEdge(7, 9, 3);
 addEdge(8, 10, 5);
 addEdge(9, 11, 2);
 addEdge(10, 12, 6);
 addEdge(11, 13, 7);
 addEdge(12, 14, 3);
 addEdge(13, 15, 2);  addEdge(14, 16, 8);  addEdge(15, 17, 4);  addEdge(16, 18, 5);
 addEdge(17, 19, 6);  addEdge(18, 20, 7);
 addEdge(19, 21, 3);  addEdge(20, 22, 2);
 addEdge(21, 23, 8);  addEdge(22, 24, 4);
 addEdge(23, 25, 5);  addEdge(24, 26, 7);
 addEdge(25, 27, 2);
addEdge(26, 28, 6);
 addEdge(27, 29, 4);
addEdge(28, 30, 5);
 addEdge(29, 31, 7);  addEdge(30, 32, 3);
 addEdge(31, 33, 6);
addEdge(32, 34, 8); addEdge(33, 35, 2);
addEdge(34, 36, 5);
 addEdge(35, 37, 4);  addEdge(36, 38, 7);
 addEdge(37, 39, 1);  addEdge(38, 39, 3);
```



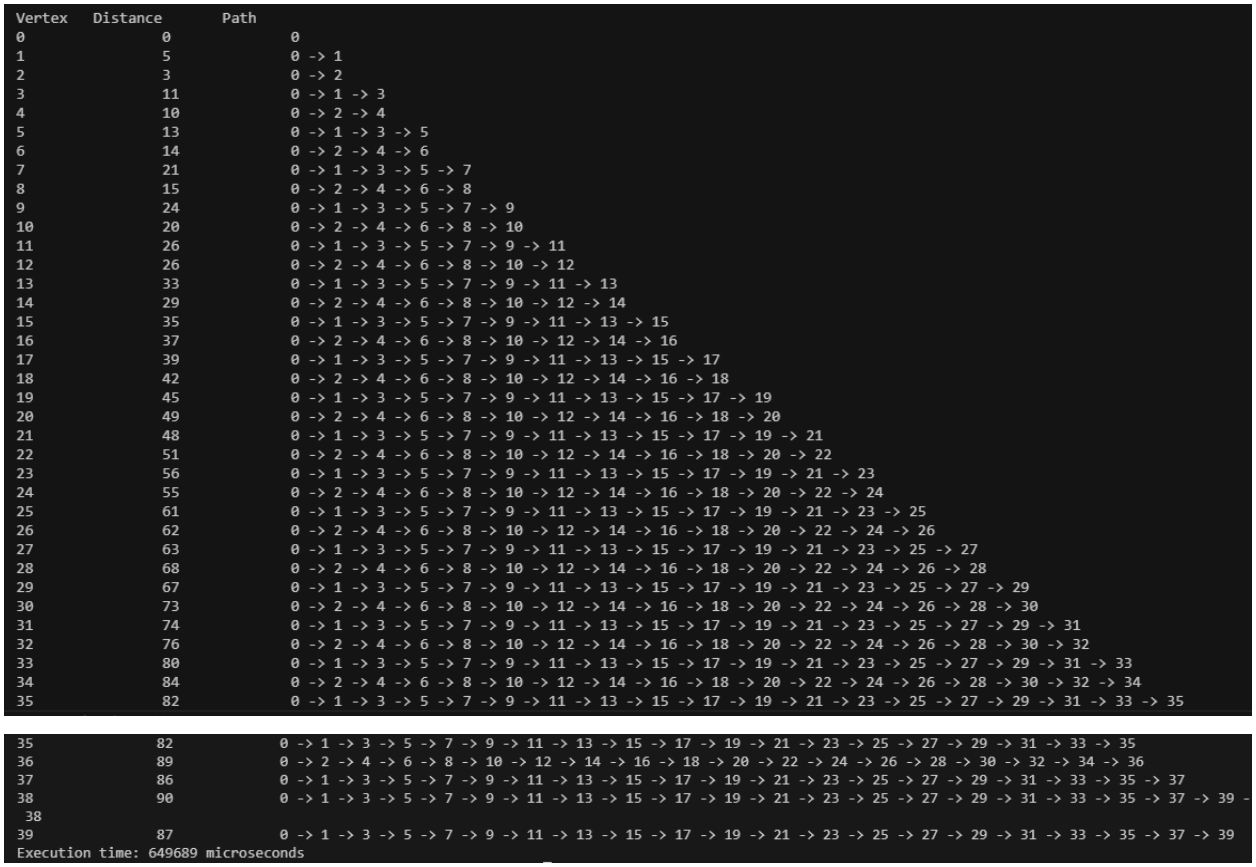
REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|2|4 (Distance: 2)
- Path to 5: 0|1|3|5 (Distance: 3)
- Path to 6: 0|2|4|6 (Distance: 3)
- Path to 7: 0|1|3|5|7 (Distance: 4)
- Path to 8: 0|2|4|6|8 (Distance: 4)
- Path to 9: 0|1|3|5|7|9 (Distance: 5)
- Path to 10: 0|2|4|6|8|10 (Distance: 5)
- Path to 11: 0|1|3|5|7|9|11 (Distance: 6)
- Path to 12: 0|2|4|6|8|10|12 (Distance: 6)
- Path to 13: 0|1|3|5|7|9|11|13 (Distance: 7)
- Path to 14: 0|2|4|6|8|10|12|14 (Distance: 7)
- Path to 15: 0|1|3|5|7|9|11|13|15 (Distance: 8)
- Path to 16: 0|2|4|6|8|10|12|14|16 (Distance: 8)
- Path to 17: 0|1|3|5|7|9|11|13|15|17 (Distance: 9)
- Path to 18: 0|2|4|6|8|10|12|14|16|18 (Distance: 9)
- Path to 19: 0|1|3|5|7|9|11|13|15|17|19 (Distance: 10)
- Path to 20: 0|2|4|6|8|10|12|14|16|18|20 (Distance: 10)
- Path to 21: 0|1|3|5|7|9|11|13|15|17|19|21 (Distance: 11)
- Path to 22: 0|2|4|6|8|10|12|14|16|18|20|22 (Distance: 11)
- Path to 23: 0|1|3|5|7|9|11|13|15|17|19|21|23 (Distance: 12)
- Path to 24: 0|2|4|6|8|10|12|14|16|18|20|22|24 (Distance: 12)
- Path to 25: 0|1|3|5|7|9|11|13|15|17|19|21|23|25 (Distance: 13)
- Path to 26: 0|2|4|6|8|10|12|14|16|18|20|22|24|26 (Distance: 13)
- Path to 27: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27 (Distance: 14)
- Path to 28: 0|2|4|6|8|10|12|14|16|18|20|22|24|26|28 (Distance: 14)
- Path to 29: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29 (Distance: 15)
- Path to 30: 0|2|4|6|8|10|12|14|16|18|20|22|24|26|28|30 (Distance: 15)
- Path to 31: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31 (Distance: 16)
- Path to 32: 0|2|4|6|8|10|12|14|16|18|20|22|24|26|28|30|32 (Distance: 16)
- Path to 33: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31|33 (Distance: 17)
- Path to 34: 0|2|4|6|8|10|12|14|16|18|20|22|24|26|28|30|32|34 (Distance: 17)
- Path to 35: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31|33|35 (Distance: 18)
- Path to 36: 0|2|4|6|8|10|12|14|16|18|20|22|24|26|28|30|32|34|36 (Distance: 18)
- Path to 37: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31|33|35|37 (Distance: 19)
- Path to 38: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31|33|35|37|39|38 (Distance: 21)
- Path to 39: 0|1|3|5|7|9|11|13|15|17|19|21|23|25|27|29|31|33|35|37|39 (Distance: 20)

CPP (Execution time: 649689 microseconds)



```
Vertex   Distance       Path
0               0              0
1               5              0 -> 1
2               3              0 -> 2
3               11             0 -> 1 -> 3
4               10             0 -> 2 -> 4
5               13             0 -> 1 -> 3 -> 5
6               14             0 -> 2 -> 4 -> 6
7               21             0 -> 1 -> 3 -> 5 -> 7
8               15             0 -> 2 -> 4 -> 6 -> 8
9               24             0 -> 1 -> 3 -> 5 -> 7 -> 9
10              20             0 -> 2 -> 4 -> 6 -> 8 -> 10
11              26             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11
12              26             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12
13              33             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13
14              29             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14
15              35             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15
16              37             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16
17              39             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17
18              42             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18
19              45             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19
20              49             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20
21              48             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21
22              51             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22
23              56             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23
24              55             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24
25              61             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25
26              62             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26
27              63             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27
28              68             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28
29              67             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29
30              73             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30
31              74             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31
32              76             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32
33              80             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33
34              84             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32 -> 34
35              82             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35
```



```
35              82             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35
36              89             0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32 -> 34 -> 36
37              86             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37
38              90             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37 -> 39 -
 38
39              87             0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37 -> 39
Execution time: 649689 microseconds
```

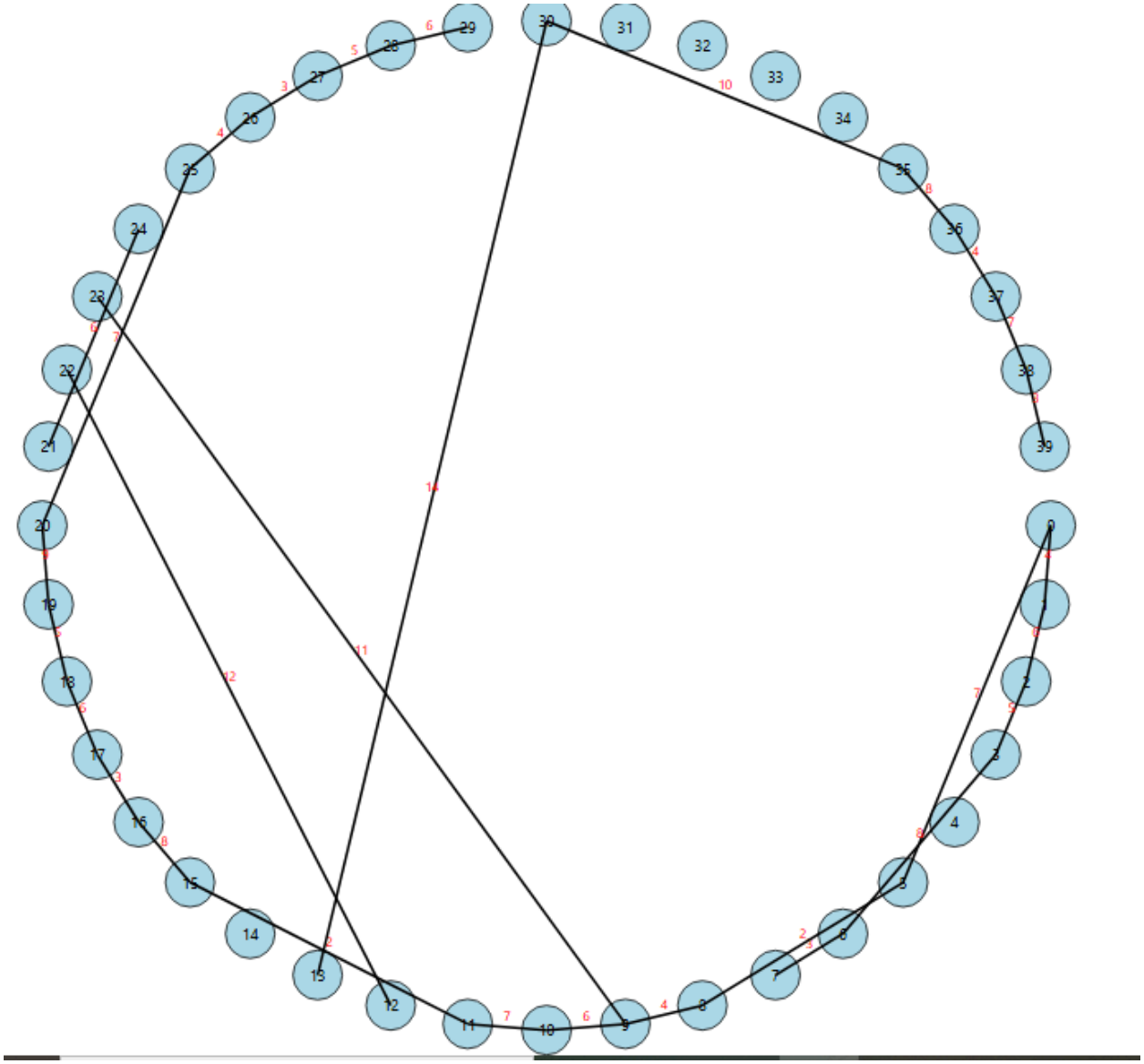
PYTHON (Execution time: 135166.700 microseconds)

```
Vertex  Distance        Path
0               0               0
1               5               0 -> 1
2               3               0 -> 2
3               11              0 -> 1 -> 3
4               10              0 -> 2 -> 4
5               13              0 -> 1 -> 3 -> 5
6               14              0 -> 2 -> 4 -> 6
8               15              0 -> 2 -> 4 -> 6 -> 8
10              20              0 -> 2 -> 4 -> 6 -> 8 -> 10
11              26              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11
12              26              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12
13              33              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13
14              29              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14
15              35              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15
16              37              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16
17              39              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17
18              42              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18
19              45              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19
20              49              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20
21              48              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21
22              51              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22
23              56              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23
24              55              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24
25              61              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25
26              62              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26
27              63              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27
28              68              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28
29              67              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29
30              73              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30
31              74              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31
32              76              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32
33              80              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33
34              84              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32 -> 34
35              82              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35
36              89              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32 -> 34 -> 36
37              86              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37
```

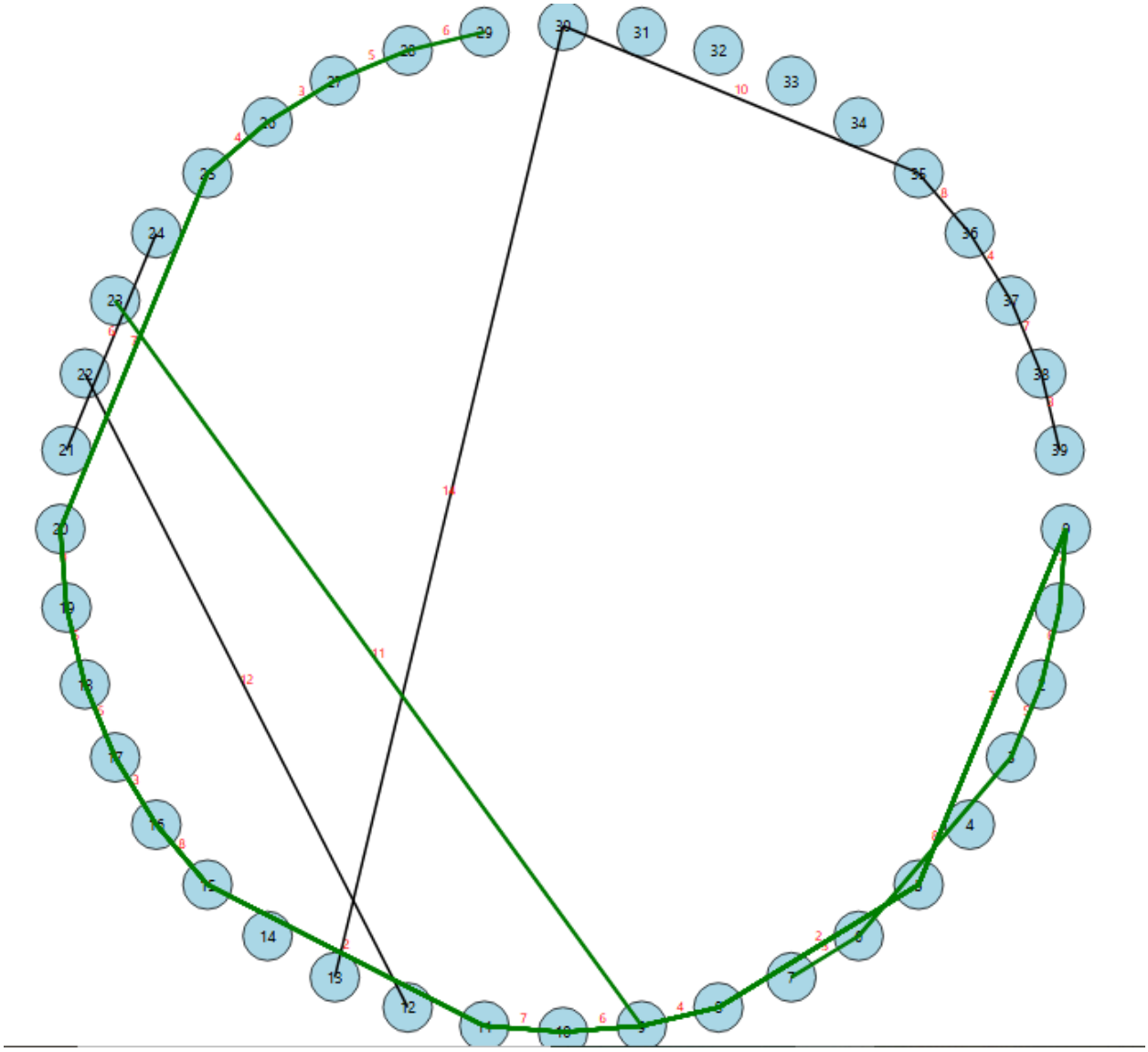
```
36              89              0 -> 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22 -> 24 -> 26 -> 28 -> 30 -> 32 -> 34 -> 36
37              86              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37
38              90              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37 -> 39 ->
 38
39              87              0 -> 1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> 33 -> 35 -> 37 -> 39
Execution time: 135166.700 microseconds
```

TEST # 5

```
addEdge(0, 1, 4);
addEdge(0, 5, 7);
addEdge(1, 2, 6);
addEdge(2, 3, 5);
addEdge(3, 6, 8);
addEdge(6, 7, 3);
addEdge(5, 8, 2);
addEdge(8, 9, 4);
addEdge(9, 10, 6);
addEdge(10, 11, 7);
addEdge(11, 15, 2);
addEdge(15, 16, 8);
addEdge(16, 17, 3);
addEdge(17, 18, 6);
addEdge(18, 19, 5);
addEdge(19, 20, 9);
addEdge(20, 25, 7);
addEdge(25, 26, 4);
addEdge(26, 27, 3);
addEdge(27, 28, 5);
addEdge(28, 29, 6);
addEdge(30, 35, 10);
addEdge(35, 36, 8);
addEdge(36, 37, 4);
addEdge(37, 38, 7);
addEdge(38, 39, 3);
addEdge(12, 22, 12);
addEdge(13, 30, 14);
addEdge(21, 24, 6);
addEdge(9, 23, 11);
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

REACT


30
31
32
33
34
35
36
37
38
39
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|1|2 (Distance: 2)
- Path to 3: 0|1|2|3 (Distance: 3)
-
- Path to 5: 0|5 (Distance: 1)
- Path to 6: 0|1|2|3|6 (Distance: 4)
- Path to 7: 0|1|2|3|6|7 (Distance: 5)
- Path to 8: 0|5|8 (Distance: 2)
- Path to 9: 0|5|8|9 (Distance: 3)
- Path to 10: 0|5|8|9|10 (Distance: 4)
- Path to 11: 0|5|8|9|10|11 (Distance: 5)
-
-
-
- Path to 15: 0|5|8|9|10|11|15 (Distance: 6)
- Path to 16: 0|5|8|9|10|11|15|16 (Distance: 7)
- Path to 17: 0|5|8|9|10|11|15|16|17 (Distance: 8)
- Path to 18: 0|5|8|9|10|11|15|16|17|18 (Distance: 9)
- Path to 19: 0|5|8|9|10|11|15|16|17|18|19 (Distance: 10)
- Path to 20: 0|5|8|9|10|11|15|16|17|18|19|20 (Distance: 11)
-
-
- Path to 23: 0|5|8|9|23 (Distance: 4)
-
- Path to 25: 0|5|8|9|10|11|15|16|17|18|19|20|25 (Distance: 12)
- Path to 26: 0|5|8|9|10|11|15|16|17|18|19|20|25|26 (Distance: 13)
- Path to 27: 0|5|8|9|10|11|15|16|17|18|19|20|25|26|27 (Distance: 14)
- Path to 28: 0|5|8|9|10|11|15|16|17|18|19|20|25|26|27|28 (Distance: 15)
- Path to 29: 0|5|8|9|10|11|15|16|17|18|19|20|25|26|27|28|29 (Distance: 16)
-
-

CPP(Execution time: 240703 microseconds)

```
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0                0              0
1                4              0 -> 1
2                10             0 -> 1 -> 2
3                15             0 -> 1 -> 2 -> 3
4                1061109567
5                7              0 -> 5
6                23             0 -> 1 -> 2 -> 3 -> 6
7                26             0 -> 1 -> 2 -> 3 -> 6 -> 7
8                9              0 -> 5 -> 8
9                13             0 -> 5 -> 8 -> 9
10               19             0 -> 5 -> 8 -> 9 -> 10
11               26             0 -> 5 -> 8 -> 9 -> 10 -> 11
12               1061109567
13               1061109567
14               1061109567
15               28             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15
16               36             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16
17               39             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17
18               45             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18
19               50             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19
20               59             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21               1061109567
22               1061109567
23               24             0 -> 5 -> 8 -> 9 -> 23
24               1061109567
25               66             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25
26               70             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26
27               73             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27
28               78             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27 -> 28
29               84             0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27 -> 28 -> 29
30               1061109567
31               1061109567
32               1061109567
33               1061109567
34               1061109567
```

```
33               1061109567
34               1061109567
35               1061109567
36               1061109567
37               1061109567
38               1061109567
39               1061109567
Execution time: 240703 microseconds
```

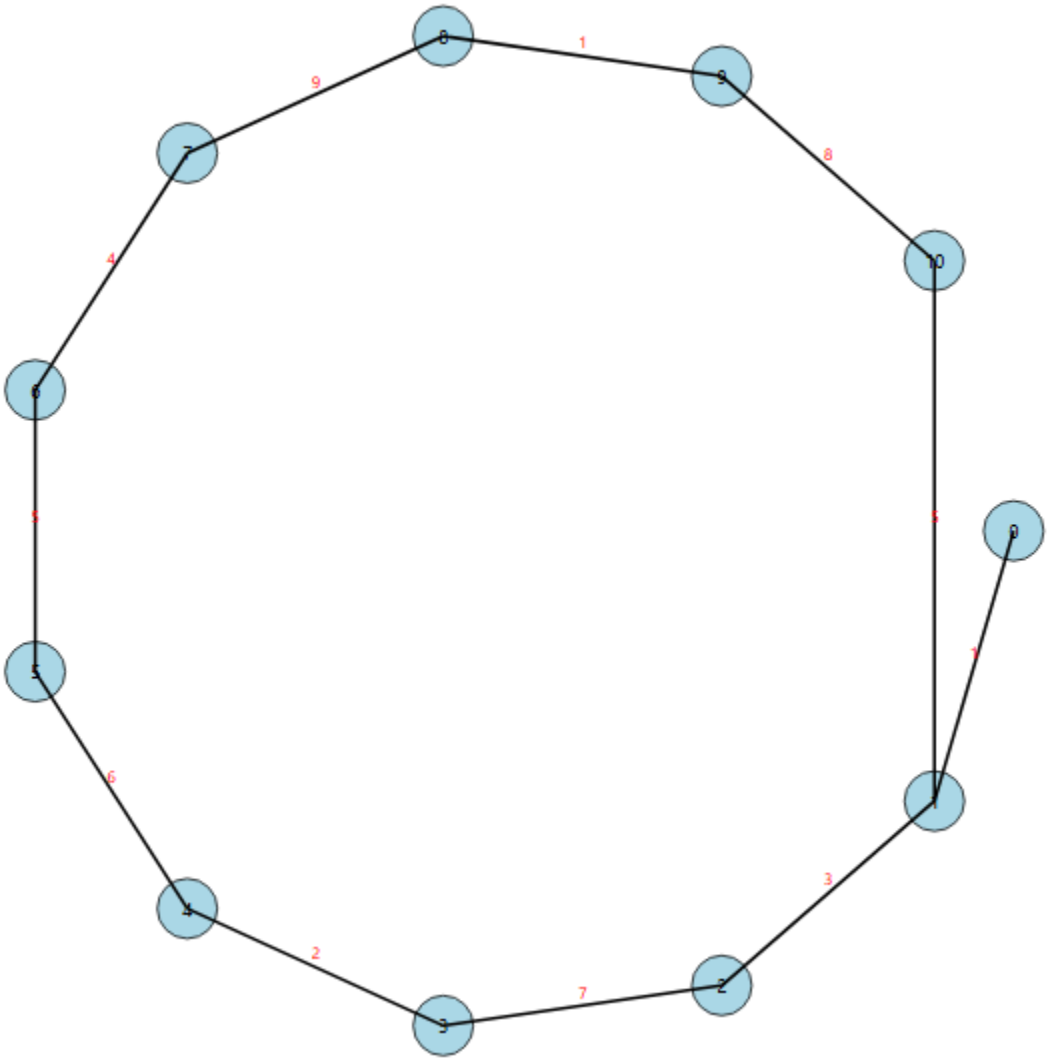
PYTHON (Execution time: 44986.200 microseconds)

```
Vertex  Distance        Path
0               0               0
1               4               0 -> 1
2               10              0 -> 1 -> 2
3               15              0 -> 1 -> 2 -> 3
4               inf
5               7               0 -> 5
6               23              0 -> 1 -> 2 -> 3 -> 6
7               26              0 -> 1 -> 2 -> 3 -> 6 -> 7
8               9               0 -> 5 -> 8
9               13              0 -> 5 -> 8 -> 9
10              19              0 -> 5 -> 8 -> 9 -> 10
11              26              0 -> 5 -> 8 -> 9 -> 10 -> 11
12              inf
13              inf
14              inf
15              28              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15
16              36              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16
17              39              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17
18              45              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18
19              50              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19
20              59              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20
21              inf
22              inf
23              24              0 -> 5 -> 8 -> 9 -> 23
24              inf
25              66              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25
26              70              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26
27              73              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27
28              78              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27 -> 28
29              84              0 -> 5 -> 8 -> 9 -> 10 -> 11 -> 15 -> 16 -> 17 -> 18 -> 19 -> 20 -> 25 -> 26 -> 27 -> 28 -> 29
30              inf
31              inf
32              inf
33              inf
34              inf
35              inf
```

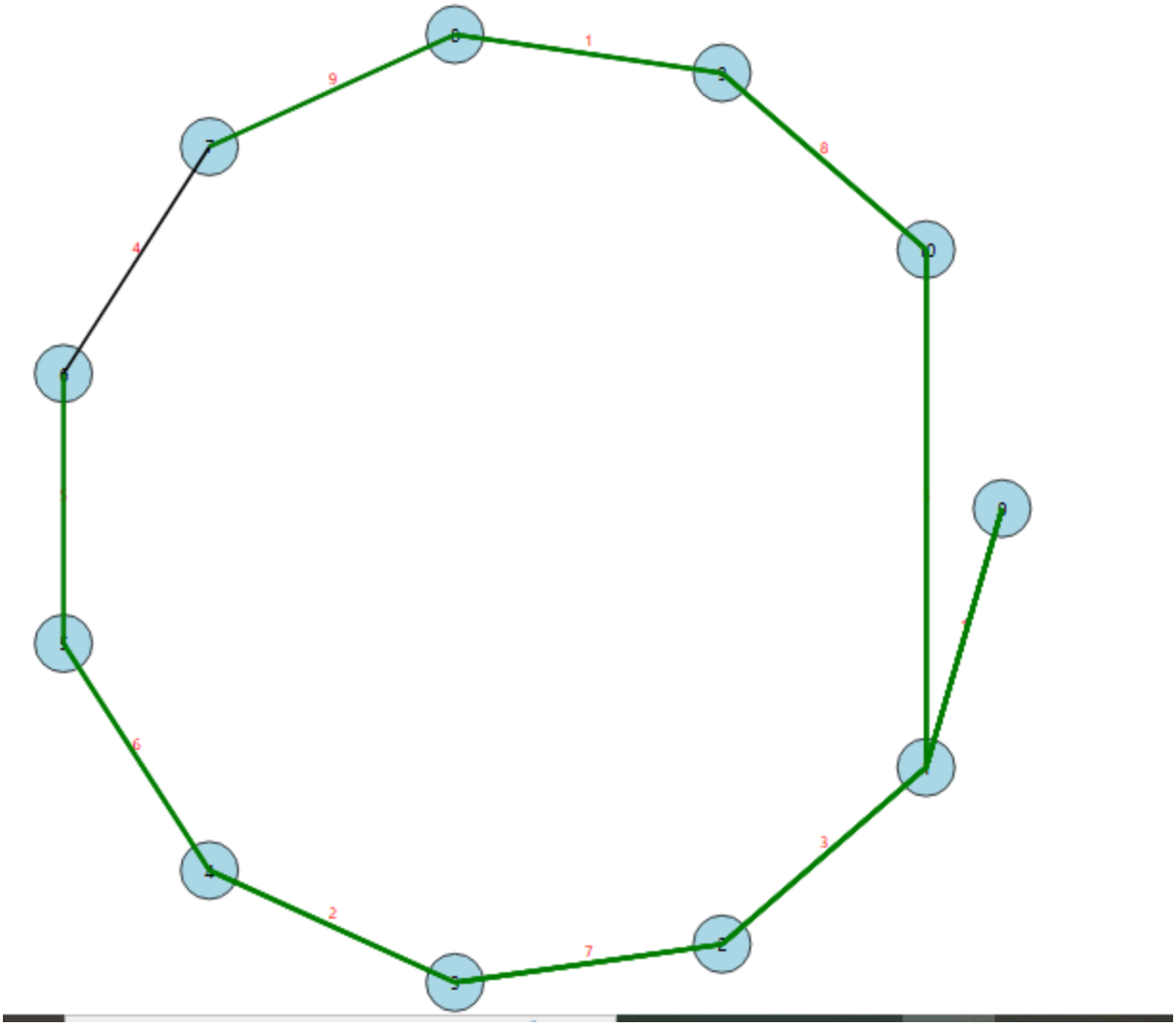
```
34              inf
35              inf
36              inf
37              inf
38              inf
39              inf
Execution time: 44986.200 microseconds
```

TEST # 6

```
g.addEdge(1, 2, 3);
g.addEdge(2, 3, 7);
g.addEdge(3, 4, 2);
g.addEdge(4, 5, 6);
g.addEdge(5, 6, 5);
g.addEdge(6, 7, 4);
g.addEdge(7, 8, 9);
g.addEdge(8, 9, 1);
g.addEdge(9, 10, 8);
g.addEdge(10, 1, 5);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|1|2 (Distance: 2)
- Path to 3: 0|1|2|3 (Distance: 3)
- Path to 4: 0|1|2|3|4 (Distance: 4)
- Path to 5: 0|1|2|3|4|5 (Distance: 5)
- Path to 6: 0|1|2|3|4|5|6 (Distance: 6)
- Path to 7: 0|1|10|9|8|7 (Distance: 5)
- Path to 8: 0|1|10|9|8 (Distance: 4)
- Path to 9: 0|1|10|9 (Distance: 3)
- Path to 10: 0|1|10 (Distance: 2)

CPP(Execution time: 32783 microseconds)

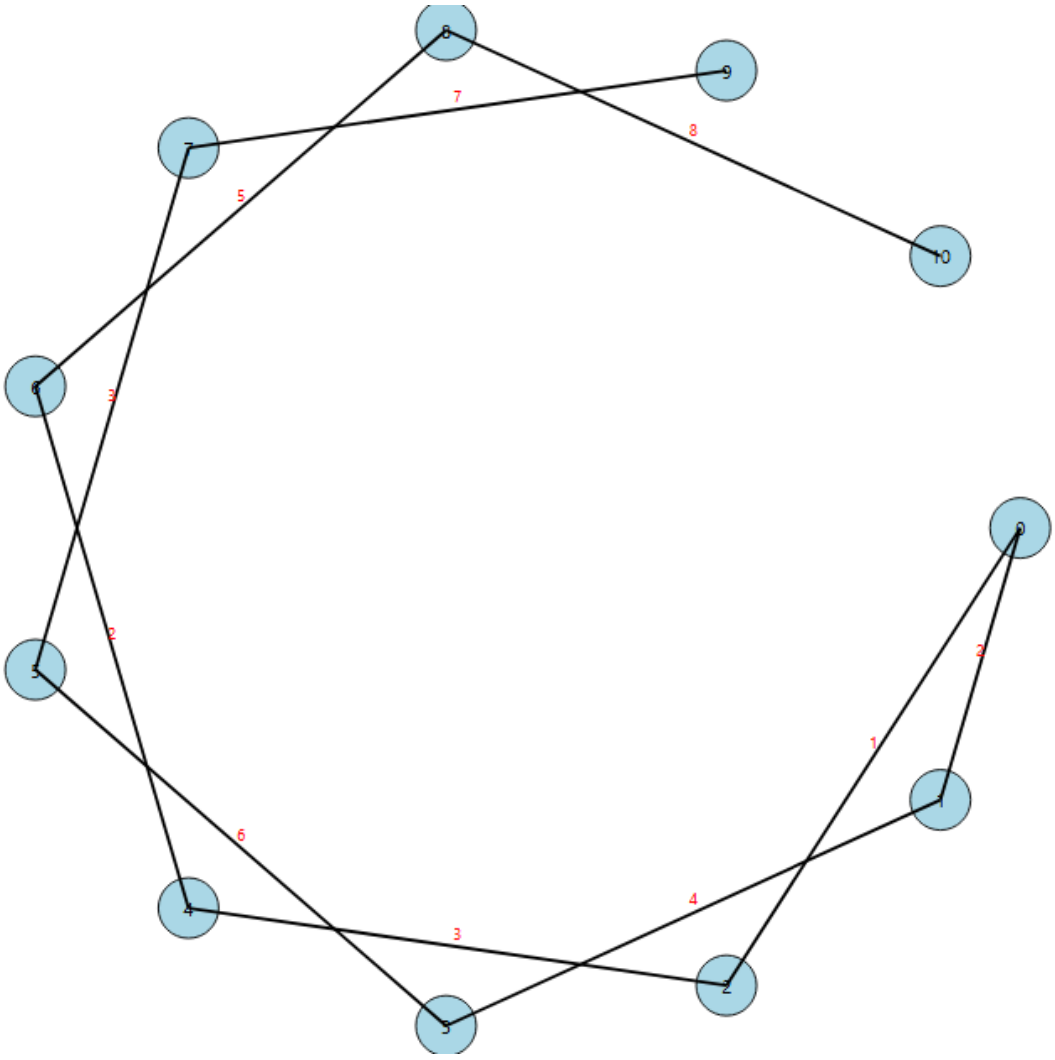
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer> cd "e:\assigment daniyal\GT PROJECT GUI\d
cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0               0               0
1               1               0 -> 1
2               4               0 -> 1 -> 2
3               11              0 -> 1 -> 2 -> 3
4               13              0 -> 1 -> 2 -> 3 -> 4
5               19              0 -> 1 -> 2 -> 3 -> 4 -> 5
6               24              0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6
7               24              0 -> 1 -> 10 -> 9 -> 8 -> 7
8               15              0 -> 1 -> 10 -> 9 -> 8
9               14              0 -> 1 -> 10 -> 9
10              6               0 -> 1 -> 10
Execution time: 32783 microseconds
```

PYTHON(Execution time: 13680.200 microseconds)

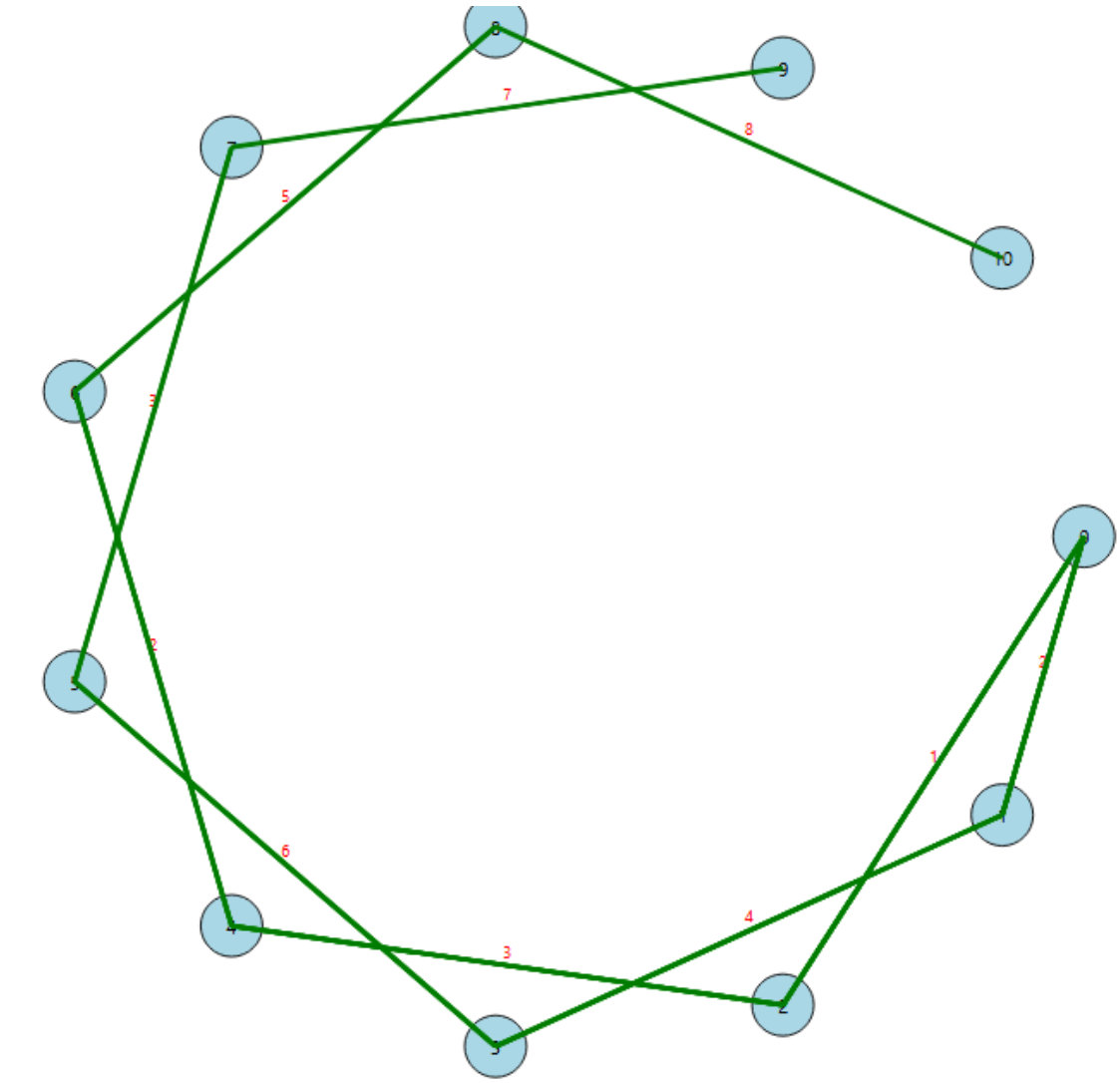
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> python -u "e:\assigment daniyal\GT PROJECT GUI
implementation.py"
Vertex  Distance        Path
0               0               0
1               1               0 -> 1
2               4               0 -> 1 -> 2
3               11              0 -> 1 -> 2 -> 3
4               13              0 -> 1 -> 2 -> 3 -> 4
5               19              0 -> 1 -> 2 -> 3 -> 4 -> 5
6               24              0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6
7               24              0 -> 1 -> 10 -> 9 -> 8 -> 7
8               15              0 -> 1 -> 10 -> 9 -> 8
9               14              0 -> 1 -> 10 -> 9
10              6               0 -> 1 -> 10
Execution time: 13680.200 microseconds
```

TEST # 7

```
g.addEdge(0, 1, 2);
g.addEdge(1, 3, 4);
g.addEdge(3, 5, 6);
g.addEdge(5, 7, 3);
g.addEdge(7, 9, 7);
g.addEdge(0, 2, 1);
g.addEdge(2, 4, 3);
g.addEdge(4, 6, 2);
g.addEdge(6, 8, 5);
g.addEdge(8, 10, 8);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|2|4 (Distance: 2)
- Path to 5: 0|1|3|5 (Distance: 3)
- Path to 6: 0|2|4|6 (Distance: 3)
- Path to 7: 0|1|3|5|7 (Distance: 4)
- Path to 8: 0|2|4|6|8 (Distance: 4)
- Path to 9: 0|1|3|5|7|9 (Distance: 5)
- Path to 10: 0|2|4|6|8|10 (Distance: 5)

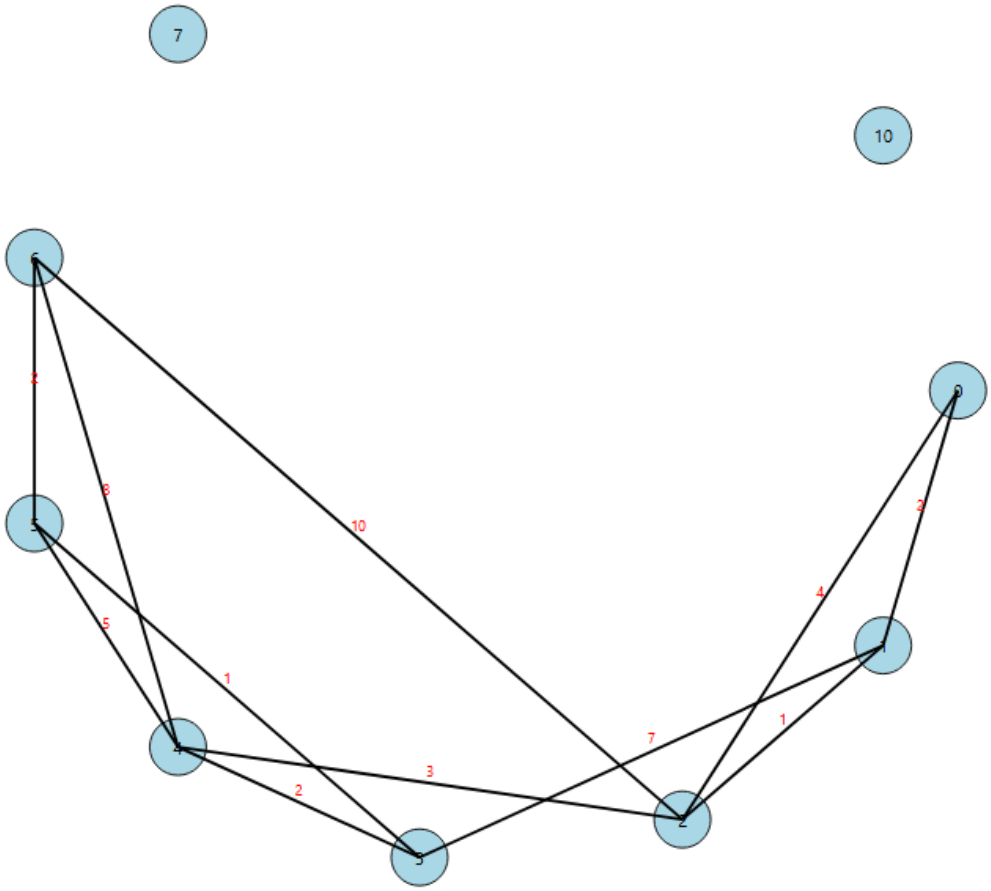
CPP (Execution time: 27506 microseconds)

```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> cd "e:\assigment d
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra
Vertex   Distance       Path
0                0              0
1                2              0 -> 1
2                1              0 -> 2
3                6              0 -> 1 -> 3
4                4              0 -> 2 -> 4
5                12             0 -> 1 -> 3 -> 5
6                6              0 -> 2 -> 4 -> 6
7                15             0 -> 1 -> 3 -> 5 -> 7
8                11             0 -> 2 -> 4 -> 6 -> 8
9                22             0 -> 1 -> 3 -> 5 -> 7 -> 9
10               19             0 -> 2 -> 4 -> 6 -> 8 -> 10
Execution time: 27506 microseconds
```

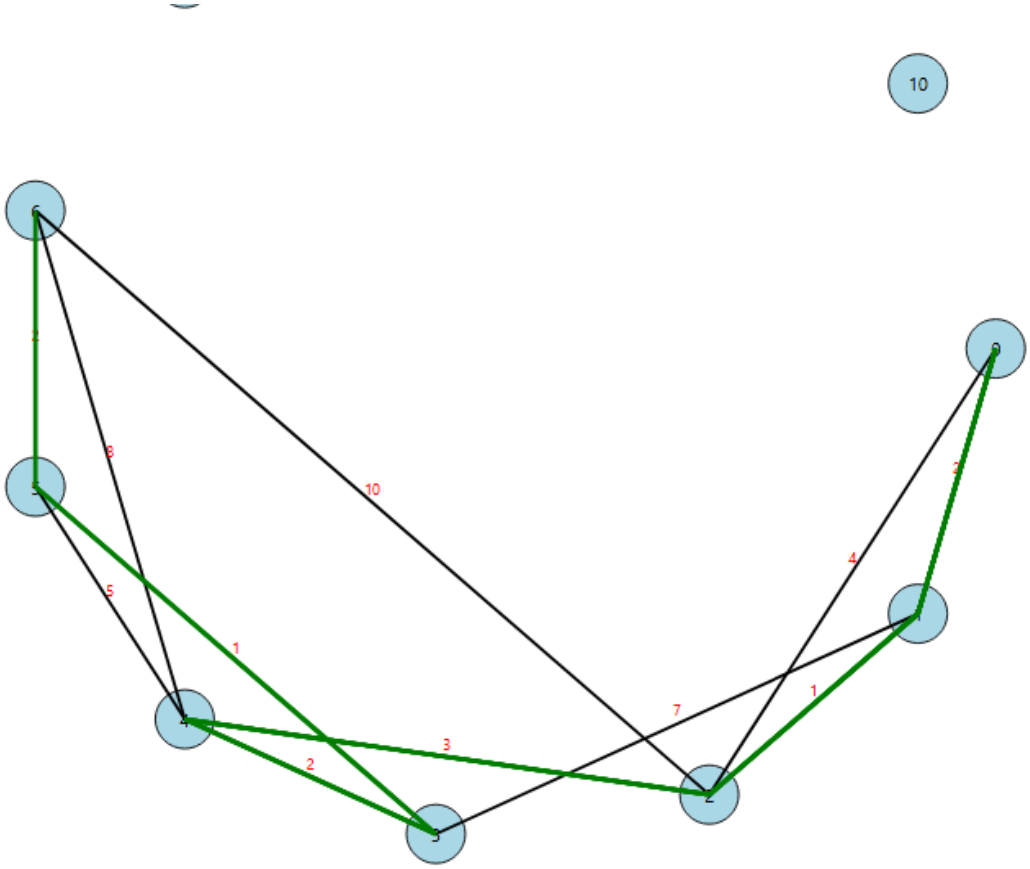
PYTHON (Execution time: 9298.600 microseconds)

TEST # 8b

```
g.addEdge(0, 1, 2);
g.addEdge(0, 2, 4);
g.addEdge(1, 2, 1);
g.addEdge(1, 3, 7);
g.addEdge(2, 4, 3);
g.addEdge(3, 4, 2);
g.addEdge(3, 5, 1);
g.addEdge(4, 5, 5);
g.addEdge(4, 6, 8);
g.addEdge(5, 6, 2);
g.addEdge(2, 6, 10);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|1|2 (Distance: 2)
- Path to 3: 0|1|2|4|3 (Distance: 4)
- Path to 4: 0|1|2|4 (Distance: 3)
- Path to 5: 0|1|2|4|3|5 (Distance: 5)
- Path to 6: 0|1|2|4|3|5|6 (Distance: 6)
-
-
-
-

CPP(Execution time: 22427 microseconds)

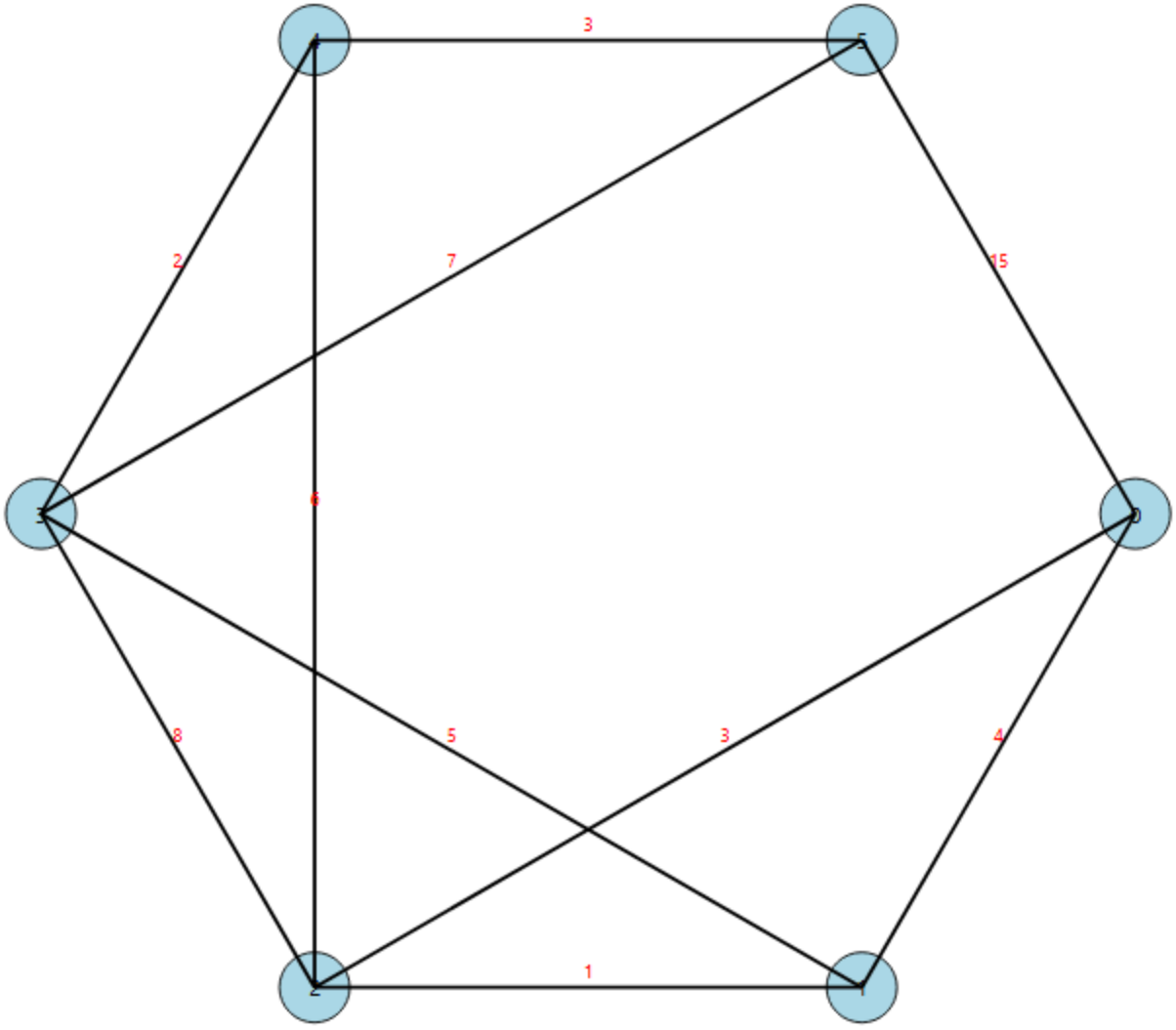
```
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_im
Vertex   Distance       Path
0                0              0
1                2              0 -> 1
2                3              0 -> 1 -> 2
3                8              0 -> 1 -> 2 -> 4 -> 3
4                6              0 -> 1 -> 2 -> 4
5                9              0 -> 1 -> 2 -> 4 -> 3 -> 5
6                11             0 -> 1 -> 2 -> 4 -> 3 -> 5 -> 6
7                1061109567
8                1061109567
9                1061109567
10               1061109567
Execution time: 22427 microseconds
```

PYTHON (Execution time: 11755.600 microseconds)

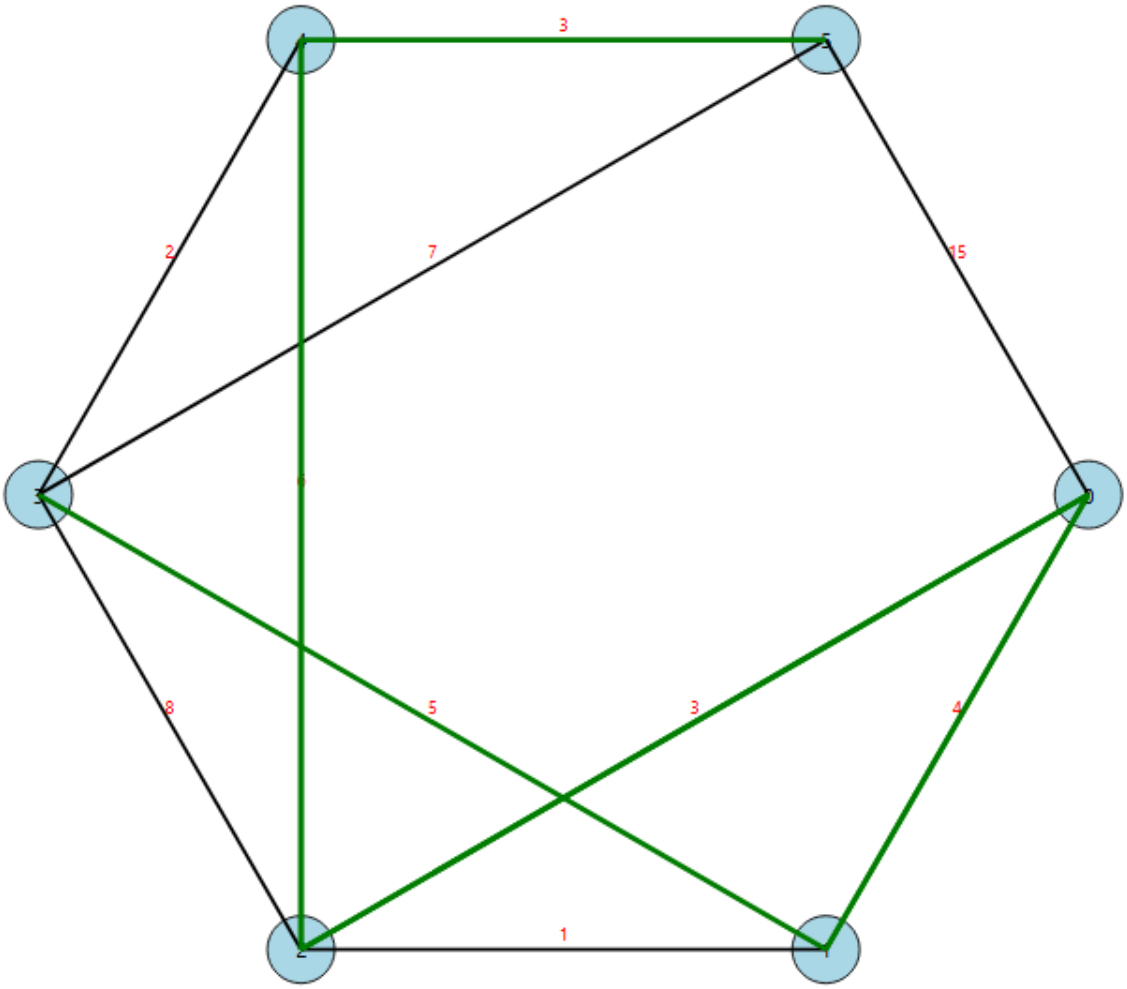
```
implementation.py"
Vertex  Distance        Path
0               0               0
1               2               0 -> 1
2               3               0 -> 1 -> 2
3               8               0 -> 1 -> 2 -> 4 -> 3
4               6               0 -> 1 -> 2 -> 4
5               9               0 -> 1 -> 2 -> 4 -> 3 -> 5
6               11              0 -> 1 -> 2 -> 4 -> 3 -> 5 -> 6
7               inf
8               inf
9               inf
10              inf
Execution time: 11755.600 microseconds
```

TEST # 8

```
 g.addEdge(0, 1, 4);
g.addEdge(0, 2, 3);
g.addEdge(1, 2, 1);
g.addEdge(1, 3, 5);
g.addEdge(2, 3, 8);
g.addEdge(2, 4, 6);
g.addEdge(3, 4, 2);
g.addEdge(3, 5, 7);
g.addEdge(4, 5, 3);
g.addEdge(0, 5, 15);
```



REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|2|4 (Distance: 2)
- Path to 5: 0|2|4|5 (Distance: 3)

CPP (Execution time: 11406 microseconds)

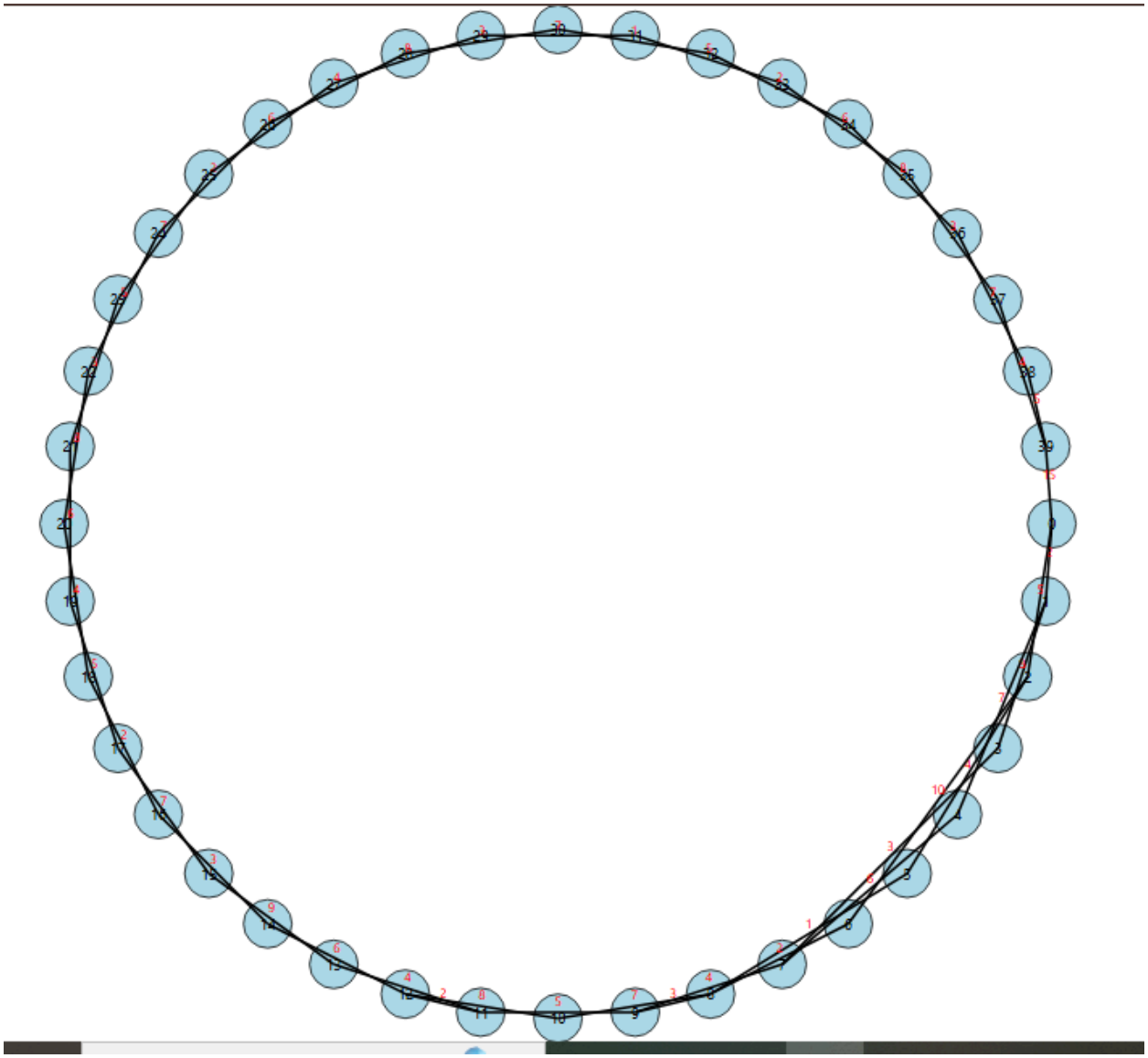
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> cd "e:\assigment daniyal\GT PROJECT GUI\
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0               0               0
1               4               0 -> 1
2               3               0 -> 2
3               9               0 -> 1 -> 3
4               9               0 -> 2 -> 4
5               12              0 -> 2 -> 4 -> 5
Execution time: 11406 microseconds
```

PYTHON (Execution time: 9848.900 microseconds)

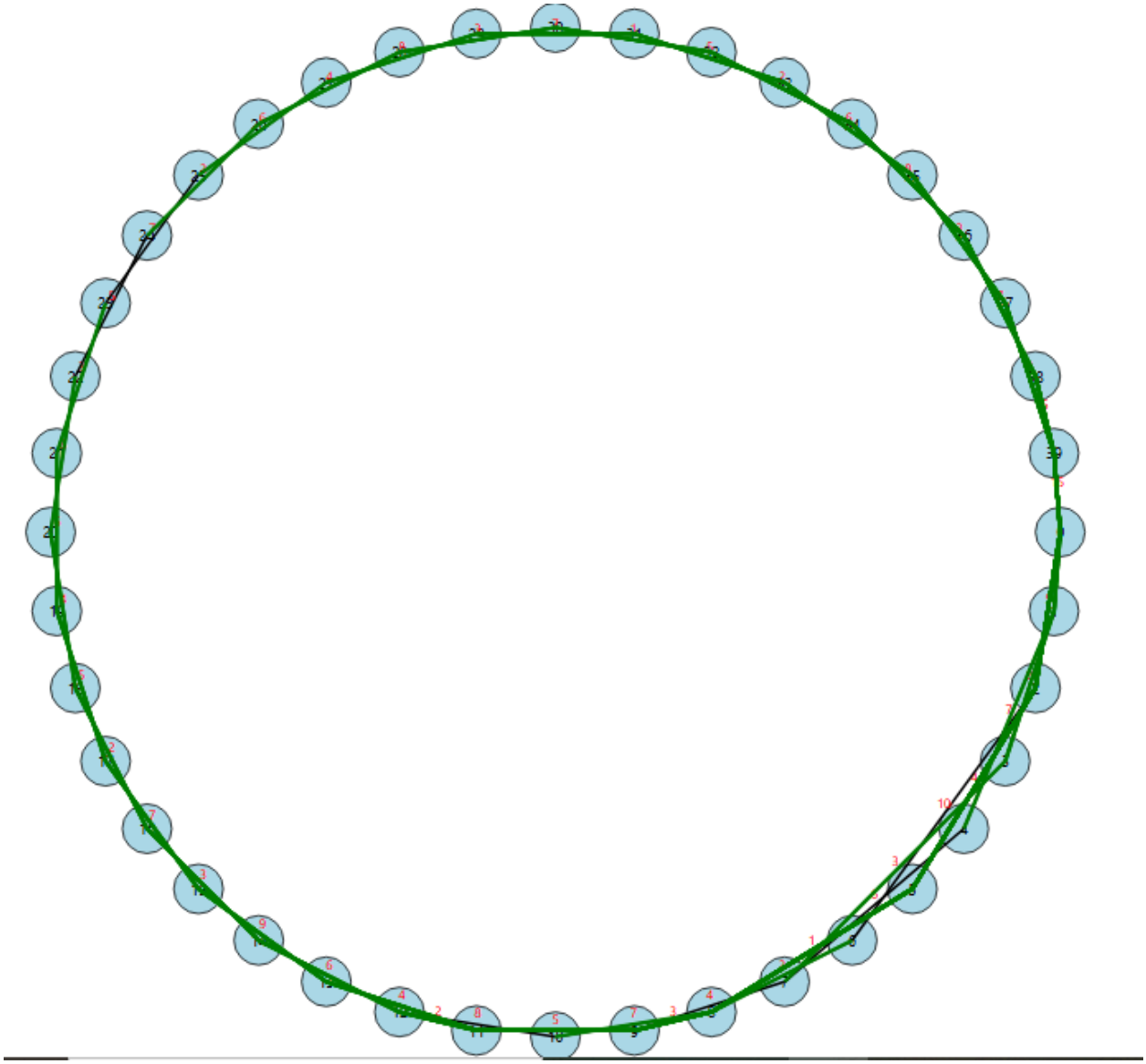
```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> python -u "e:
implementation.py"
Vertex  Distance        Path
0               0               0
1               4               0 -> 1
2               3               0 -> 2
3               9               0 -> 1 -> 3
4               9               0 -> 2 -> 4
5               12              0 -> 2 -> 4 -> 5
Execution time: 9848.900 microseconds
```

TEST # 9

```
addEdge(0, 1, 2);
addEdge(0, 2, 5);  addEdge(1, 3, 8);
addEdge(1, 4, 7);  addEdge(2, 5, 4);
addEdge(2, 6, 10);  addEdge(3, 7, 3);
addEdge(4, 7, 6);  addEdge(5, 8, 1);
addEdge(6, 8, 2);  addEdge(7, 9, 4);
addEdge(8, 9, 3);  addEdge(8, 10, 7);
addEdge(9, 11, 5);  addEdge(10, 12, 8);
addEdge(11, 12, 2);  addEdge(11, 13, 4);
addEdge(12, 14, 6);  addEdge(13, 15, 9);
addEdge(14, 16, 3); addEdge(15, 17, 7);
addEdge(16, 18, 2); addEdge(17, 19, 5);
addEdge(18, 20, 4); addEdge(19, 21, 6);
addEdge(20, 22, 8); addEdge(21, 23, 3);
addEdge(22, 24, 5); addEdge(23, 25, 7);
addEdge(24, 26, 2); addEdge(25, 27, 6);
addEdge(26, 28, 4); addEdge(27, 29, 8);
addEdge(28, 30, 3); addEdge(29, 31, 7);
addEdge(30, 32, 1); addEdge(31, 33, 5);
addEdge(32, 34, 2); addEdge(33, 35, 6);
addEdge(34, 36, 8); addEdge(35, 37, 3);
addEdge(36, 38, 7); addEdge(37, 39, 4);
addEdge(38, 39, 5); addEdge(0, 39, 15);
```



REACT



- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|1|3 (Distance: 2)
- Path to 4: 0|1|4 (Distance: 2)
- Path to 5: 0|2|5 (Distance: 2)
- Path to 6: 0|2|5|8|6 (Distance: 4)
- Path to 7: 0|1|3|7 (Distance: 3)
- Path to 8: 0|2|5|8 (Distance: 3)
- Path to 9: 0|2|5|8|9 (Distance: 4)
- Path to 10: 0|2|5|8|10 (Distance: 4)
- Path to 11: 0|2|5|8|9|11 (Distance: 5)
- Path to 12: 0|2|5|8|9|11|12 (Distance: 6)
- Path to 13: 0|2|5|8|9|11|13 (Distance: 6)
- Path to 14: 0|2|5|8|9|11|12|14 (Distance: 7)
- Path to 15: 0|2|5|8|9|11|13|15 (Distance: 7)
- Path to 16: 0|2|5|8|9|11|12|14|16 (Distance: 8)
- Path to 17: 0|2|5|8|9|11|13|15|17 (Distance: 8)
- Path to 18: 0|2|5|8|9|11|12|14|16|18 (Distance: 9)
- Path to 19: 0|2|5|8|9|11|13|15|17|19 (Distance: 9)
- Path to 20: 0|2|5|8|9|11|12|14|16|18|20 (Distance: 10)
- Path to 21: 0|2|5|8|9|11|13|15|17|19|21 (Distance: 10)
- Path to 22: 0|2|5|8|9|11|12|14|16|18|20|22 (Distance: 11)
- Path to 23: 0|2|5|8|9|11|13|15|17|19|21|23 (Distance: 11)
- Path to 24: 0|39|38|36|34|32|30|28|26|24 (Distance: 9)
- Path to 25: 0|39|37|35|33|31|29|27|25 (Distance: 8)
- Path to 26: 0|39|38|36|34|32|30|28|26 (Distance: 8)
- Path to 27: 0|39|37|35|33|31|29|27 (Distance: 7)
- Path to 28: 0|39|38|36|34|32|30|28 (Distance: 7)
- Path to 29: 0|39|37|35|33|31|29 (Distance: 6)
- Path to 30: 0|39|38|36|34|32|30 (Distance: 6)
- Path to 31: 0|39|37|35|33|31 (Distance: 5)
- Path to 32: 0|39|38|36|34|32 (Distance: 5)
- Path to 33: 0|39|37|35|33 (Distance: 4)
- Path to 34: 0|39|38|36|34 (Distance: 4)
- Path to 35: 0|39|37|35 (Distance: 3)
- Path to 36: 0|39|38|36 (Distance: 3)
- Path to 37: 0|39|37 (Distance: 2)
- Path to 38: 0|39|38 (Distance: 2)
- Path to 39: 0|39 (Distance: 1)

CPP(Execution time: 190365 microseconds)

```
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if ($?) { .\dijkstra_cpp_implementation }
Vertex   Distance       Path
0                0              0
1                2              0 -> 1
2                5              0 -> 2
3                10             0 -> 1 -> 3
4                9              0 -> 1 -> 4
5                9              0 -> 2 -> 5
6                12             0 -> 2 -> 5 -> 8 -> 6
7                13             0 -> 1 -> 3 -> 7
8                10             0 -> 2 -> 5 -> 8
9                13             0 -> 2 -> 5 -> 8 -> 9
10               17             0 -> 2 -> 5 -> 8 -> 10
11               18             0 -> 2 -> 5 -> 8 -> 9 -> 11
12               20             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12
13               22             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13
14               26             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14
15               31             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15
16               29             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16
17               38             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17
18               31             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18
19               43             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19
20               35             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18 -> 20
21               49             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21
22               43             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22
23               52             0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23
24               47             0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28 -> 26 -> 24
25               54             0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29 -> 27 -> 25
26               45             0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28 -> 26
27               48             0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29 -> 27
28               41             0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28
29               40             0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29
30               38             0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30
31               33             0 -> 39 -> 37 -> 35 -> 33 -> 31
32               37             0 -> 39 -> 38 -> 36 -> 34 -> 32
33               28             0 -> 39 -> 37 -> 35 -> 33
34               35             0 -> 39 -> 38 -> 36 -> 34
```

```
33               28             0 -> 39 -> 37 -> 35 -> 33
34               35             0 -> 39 -> 38 -> 36 -> 34
35               22             0 -> 39 -> 37 -> 35
36               27             0 -> 39 -> 38 -> 36
37               19             0 -> 39 -> 37
38               20             0 -> 39 -> 38
39               15             0 -> 39
Execution time: 190365 microseconds
```

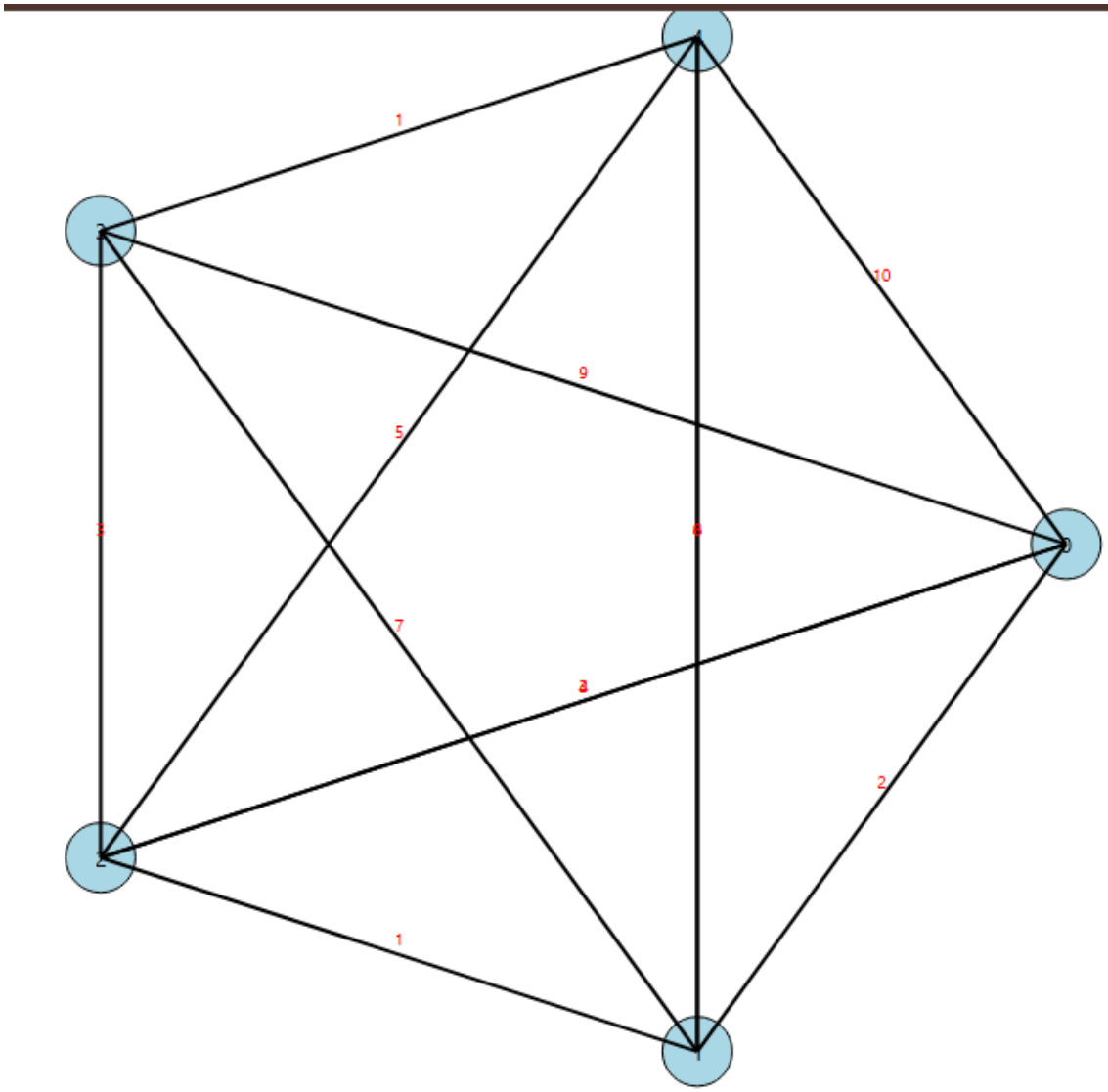
PYTHON (Execution time: 38823.100 microseconds)

```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> python -u "e:\assigment daniyal\GT PROJECT GUI\
implementation.py"
Vertex  Distance        Path
0               0               0
1               2               0 -> 1
2               5               0 -> 2
3               10              0 -> 1 -> 3
4               9               0 -> 1 -> 4
5               9               0 -> 2 -> 5
6               12              0 -> 2 -> 5 -> 8 -> 6
7               13              0 -> 1 -> 3 -> 7
8               10              0 -> 2 -> 5 -> 8
9               13              0 -> 2 -> 5 -> 8 -> 9
10              17              0 -> 2 -> 5 -> 8 -> 10
11              18              0 -> 2 -> 5 -> 8 -> 9 -> 11
12              20              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12
13              22              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13
14              26              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14
15              31              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15
16              29              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16
17              38              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17
18              31              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18
19              43              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19
20              35              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18 -> 20
21              49              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21
22              43              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 12 -> 14 -> 16 -> 18 -> 20 -> 22
23              52              0 -> 2 -> 5 -> 8 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23
24              47              0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28 -> 26 -> 24
25              54              0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29 -> 27 -> 25
26              45              0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28 -> 26
27              48              0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29 -> 27
28              41              0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30 -> 28
29              40              0 -> 39 -> 37 -> 35 -> 33 -> 31 -> 29
30              38              0 -> 39 -> 38 -> 36 -> 34 -> 32 -> 30
31              33              0 -> 39 -> 37 -> 35 -> 33 -> 31
32              37              0 -> 39 -> 38 -> 36 -> 34 -> 32
33              28              0 -> 39 -> 37 -> 35 -> 33
```

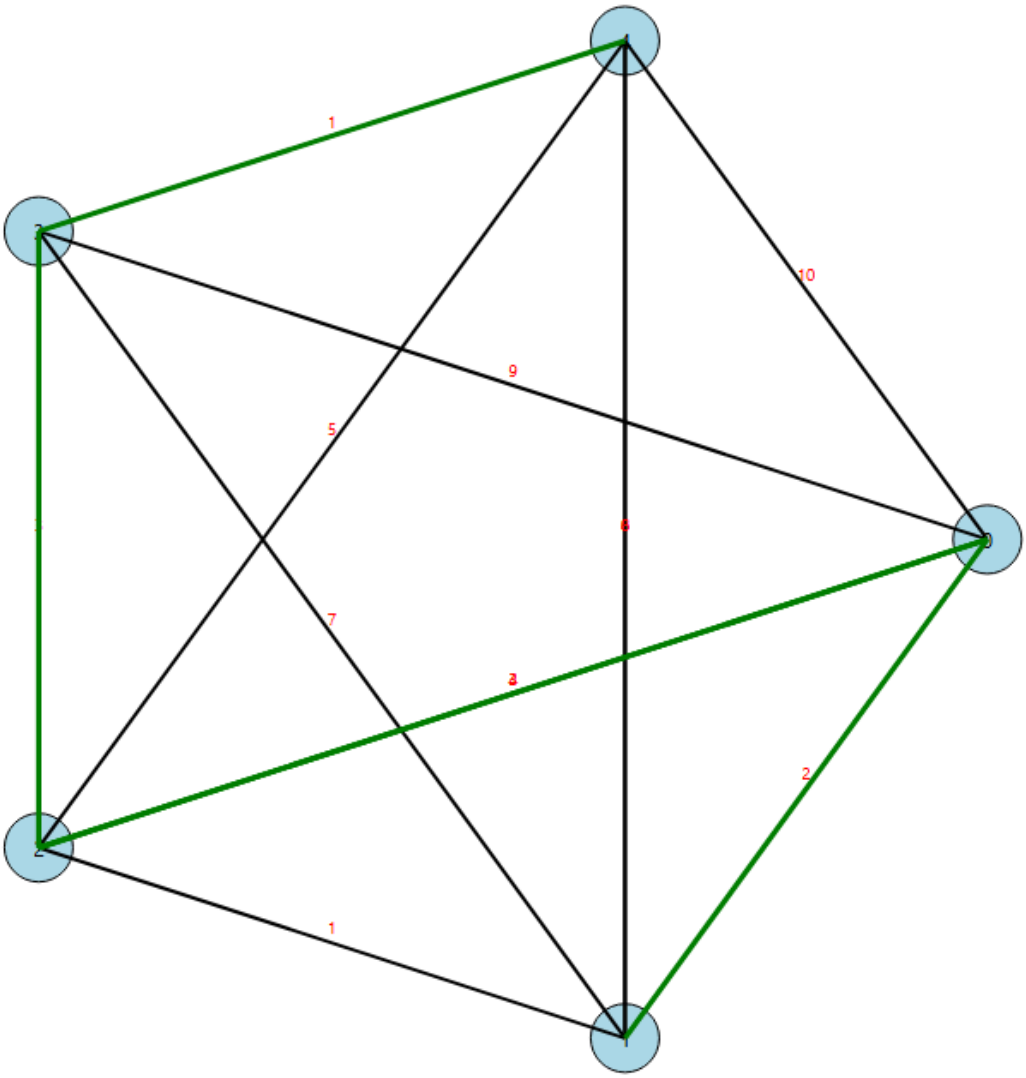
```
32              37              0 -> 39 -> 38 -> 36 -> 34 -> 32
33              28              0 -> 39 -> 37 -> 35 -> 33
34              35              0 -> 39 -> 38 -> 36 -> 34
35              22              0 -> 39 -> 37 -> 35
36              27              0 -> 39 -> 38 -> 36
37              19              0 -> 39 -> 37
38              20              0 -> 39 -> 38
39              15              0 -> 39
Execution time: 38823.100 microseconds
```

TEST # 10

```
# Adding edges with weights
g.addEdge(0, 1, 2)
g.addEdge(0, 2, 4)
g.addEdge(1, 2, 1)
g.addEdge(1, 3, 7)
g.addEdge(2, 3, 3)
g.addEdge(3, 4, 1)
g.addEdge(0, 4, 10)
g.addEdge(0, 3, 9)
g.addEdge(1, 4, 6)
g.addEdge(2, 4, 5)
g.addEdge(2, 0, 3)
g.addEdge(4, 1, 8)
```


1
10
9
5
3
6
7
8
2
1

REACT



Shortest Paths from Vertex 0

- Path to 0: 0 (Distance: 0)
- Path to 1: 0|1 (Distance: 1)
- Path to 2: 0|2 (Distance: 1)
- Path to 3: 0|2|3 (Distance: 2)
- Path to 4: 0|2|3|4 (Distance: 3)

CPP(Execution time: 8209 microseconds)

```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src> co
tra_cpp_implementation.cpp -o dijkstra_cpp_implementation } ; if (
Vertex   Distance       Path
0               0               0
1               2               0 -> 1
2               3               0 -> 2
3               6               0 -> 2 -> 3
4               7               0 -> 2 -> 3 -> 4
Execution time: 8209 microseconds
```

PYTHON (Execution time: 4287.200 microseconds)

```
PS E:\assigment daniyal\GT PROJECT GUI\dijkstra-visualizer\src
implementation.py"
Vertex  Distance        Path
0               0               0
1               2               0 -> 1
2               3               0 -> 2
3               6               0 -> 2 -> 3
4               7               0 -> 2 -> 3 -> 4
Execution time: 4287.200 microseconds
```