Project 3: Routing Algorithms

Dijkstra Algorithm Results:

Bellman-Ford Algorithm Results:

```
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test1.txt
Bellman-Ford:
[0, 1, 3, 6]
[None, 0, 1, 2]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test2.txt
Bellman-Ford:
[0, 2, 4, inf, inf]
[None, 0, 1, None, None]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test3.txt
Bellman-Ford:
[0, 6, 7, 11, 9]
[None, 0, 0, 1, 2]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test4.txt
Bellman-Ford:
None
None
[1, 2]
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test5.txt
Bellman-Ford:
[0, 5, 7]
[None, 0, 1]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test6.txt
Bellman-Ford:
[0, 7, 9, 20, 26, 11]
[None, 0, 0, 2, 3, 2]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test7.txt
Bellman-Ford:
[0, 1, 2, 3, 5]
[None, 2, 0, 1, 3]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test8.txt
Bellman-Ford:
[0, -2, inf, inf]
[None, 0, None, None]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test9.txt
Bellman-Ford:
None
None
[0]
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford> python .\bellmanford.py
Enter test case file name
test10.txt
Bellman-Ford:
[0, 3, 6, 7, 3, 12, 5, 8]
[None, 0, 0, 1, 3, 3, 4, 6]
None
PS C:\Users\nlagg\Documents\Senior Year\Computer Networks\dijkstraANDbellman-ford>
```

Results and Observations:

Dijkstra's algorithm returns the distances from the source to each node and their predecessors. Bellman-Ford's algorithm returns the distance from the source to each node, their predecessors, and a negative cycle. If there is no nonnegative cycle, then it is None. If a negative cycle is reachable from the source, then the distance and predecessors are both None. These results show the limitations of Dijkstra's algorithm. For example, when there are negatively weighted edges, Dijkstra's algorithm fails. This is prevalent in tests 3, 4, 7, 8, 9, and 10. Since Dijkstra does not revisit any nodes, it assumes that once a path is already found to a node, it is the shortest. Bellman-Ford's algorithm works with negative weights but is more computationally expensive. It calculates the shortest path using as many edges as needed. The Bellman-Ford algorithm is a distance vector algorithm and can have other issues. Some of which are the counting to infinity problem and black holing.