HW8 REPORT EREN TORLAK 210104004090 CSE222

Dijkstra's Algorithm:
Dijkstra's Algorithm is a graph search algorithm used to find the shortest path between a source vertex and all other vertices in a weighted graph. We made every edge weighted 1.

Breadth-First Search (BFS):
Breadth-First Search is a graph traversal algorithm that explores all the vertices of a graph in a breadth-first manner. Starting from a source vertex, BFS visits all the neighbors of the source vertex before moving on to their neighbors. It uses a queue data structure to keep track of the vertices to be visited.

Dijkstra's Algorithm:

Map08.txt: 9450 ms
Map09.txt: 9243 ms
Map06.txt: 9826 ms
Map10.txt: 9576 ms
Map03.txt: 9829 ms
Map05.txt: 9591 ms
Map07.txt: 9188 ms
Map02.txt: 9609 ms
Map01.txt: 5782 ms
Map04.txt: 5872 ms
pisa.txt: 2642 ms
triumph.txt: 3083 ms
tokyo.txt: 2960 ms
vatican.txt: 3446 ms
BFS:

Map08.txt: 406 ms
Map09.txt: 424 ms
Map06.txt: 506 ms
Map10.txt: 494 ms
Map03.txt: 495 ms
Map05.txt: 500 ms
Map07.txt: 510 ms
Map02.txt: 461 ms
Map01.txt: 475 ms
Map04.txt: 441 ms
pisa.txt: 519 ms
triumph.txt: 568 ms
tokyo.txt: 575 ms
vatican.txt: 493 ms
Dijkstra's Algorithm generally takes more time to compute the shortest paths compared to BFS. This is expected since Dijkstra's Algorithm has a time complexity of O((V + E) log V) while BFS has a time complexity of O(V + E).

HW8 REPORT EREN TORLAK 210104004090 CSE222

Complexity Analysis:

1. *Dijkstra's Algorithm:*

Heap as the priority queue, the time complexity is O((V + E) log V), where V is the number of vertices and E is the number of edges.

2. *BFS:*

The time complexity of BFS is O(V + E)

Summary :
Dijkstra's Algorithm has a higher time complexity than BFS due to the additional overhead of maintaining a priority queue. However, it guarantees the shortest path from the source to all other vertices.
BFS has a lower time complexity but only provides the shortest path from the source to a single destination.
Dijkstra's Algorithm performs well for finding single-source shortest paths, while BFS is efficient for finding shortest paths from a single source to all reachable vertices.