Assignment 3 report

Jesper Wingren
Jw223rn@student.lnu.se
1dt907

Shortest path results

The shortest path problem is tested by two different algorithms, Dijkstra, and Bellman-Ford. Using my timeit class I test the time for different number of vertices and edges to see how each algorithm performs.

As seen from the results below the Dijkstra becomes faster with more vertices for the same number of edges which can see to be wrong but because it uses a priority queue dense graphs make the algorithm slower. Why Dijkstra's gets slower when having a dense graph is because the priority queue can have a time complexity of $O(V^2)$ because it is repeating itself. Although because it uses a priority queue, and the Bellman-Ford doesn't it is still much faster and the more edges and vertices the bigger the difference because of Bellman-Ford using a sort of brute force to solve the problem. The negative thing with Dijkstra is that it can't handle negative graphs hence Bellman-Ford even if it's slow can be useful in some special cases.

```
Timing with 100 vertices:
5000 edges:
Dijkstra Algorithm: 6.563166 Milliseconds
Bellman-Ford Algorithm: 7.151083 Milliseconds
10000 edges:
Dijkstra Algorithm: 2.085 Milliseconds
Bellman-Ford Algorithm: 5.471541 Milliseconds
20000 edges:
Dijkstra Algorithm: 1.453125 Milliseconds
Bellman-Ford Algorithm: 25.207083 Milliseconds
30000 edges:
Dijkstra Algorithm: 2.083375 Milliseconds
Bellman-Ford Algorithm: 4.729708 Milliseconds

Timing with 500 vertices:
5000 edges:
Dijkstra Algorithm: 0.721667 Milliseconds
Bellman-Ford Algorithm: 6.343917 Milliseconds
10000 edges:
Dijkstra Algorithm: 1.104458 Milliseconds
Bellman-Ford Algorithm: 11.674125 Milliseconds
20000 edges:
Dijkstra Algorithm: 1.654333 Milliseconds
Bellman-Ford Algorithm: 21.125458 Milliseconds
30000 edges:
Dijkstra Algorithm: 1.435375 Milliseconds
Bellman-Ford Algorithm: 30.032917 Milliseconds
```

```
Timing with 1000 vertices:
5000 edges:
Dijkstra Algorithm: 0.762 Milliseconds
Bellman-Ford Algorithm: 22.107042 Milliseconds
10000 edges:
Dijkstra Algorithm: 1.08875 Milliseconds
Bellman-Ford Algorithm: 29.25425 Milliseconds
20000 edges:
Dijkstra Algorithm: 1.6235 Milliseconds
Bellman-Ford Algorithm: 50.290125 Milliseconds
30000 edges:
Dijkstra Algorithm: 2.844834 Milliseconds
Bellman-Ford Algorithm: 73.601959 Milliseconds

Timing with 5000 vertices:
5000 edges:
Dijkstra Algorithm: 0.019416 Milliseconds
Bellman-Ford Algorithm: 313.874333 Milliseconds
10000 edges:
Dijkstra Algorithm: 1.012083 Milliseconds
Bellman-Ford Algorithm: 360.092708 Milliseconds
20000 edges:
Dijkstra Algorithm: 2.395209 Milliseconds
Bellman-Ford Algorithm: 480.894833 Milliseconds
30000 edges:
Dijkstra Algorithm: 2.358458 Milliseconds
Bellman-Ford Algorithm: 635.805084 Milliseconds
```