

# CS141 – Intermediate Algorithms and Data Structures

## Assignment 2 – All Pairs Shortest Path

[Tianyou Xu]

### Abstract

## 1 Introduction

- What is the problem that you are solving?  
Finding the shortest path between different pairs in a given graph with  $E$  edges and  $V$  vertices.
- What methods are you going to use to solve the problem?  
The Bellman-Ford and the Floyd-Warshall.
- Why are these good methods to use?  
These two algorithms can handle negative edges while the Dijkstra's algorithm can't. Also these two algorithms are effective.
- Why are you going to be using both of them?  
Compare and contrast makes thing be acceptable and reliable.

## 2 Bellman-Ford

- What is the Bellman-Ford algorithm?  
The BellmanFord algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph.
- Why are you using it?  
To find the shortest distance between a given vertex to all other vertices in a graph.

- How did you adapt it to work for all-pairs as opposed to single source?  
Use loop to iterate all vertices in graph and apply Bellman-Ford algorithm.

- What is the run-time of the algorithm before and after your adaptation?  
Before:  $O(|V| * |E|)$ . After:  $O(|V|^2 * |E|)$

## 3 Floyd-Warshall

- What is the Floyd-Warshall algorithm?  
In computer science, the FloydWarshall algorithm is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles)
- Why are you using it?  
Finding the result and compare the runtime with Bellman-Ford.
- How is it better than the Bellman-Ford algorithm?  
We don't need any adapting for using Floyd-Warshall algorithm.
- What is the run-time of the algorithm?  
 $O(|V|^3)$

## 4 Results

- Compare and contrast the two algorithms?  
What makes one more suited for this problem?  
The Floyd-Warshall algorithm.

| Benchmarks | Bellman-Ford     |          | Floyd-Warshall |          |
|------------|------------------|----------|----------------|----------|
|            | $O(\cdot)$       | Actual   | $O(\cdot)$     | Actual   |
| input1.txt | $O( V ^2 *  E )$ | 0.000809 | $O( V ^3)$     | 0.000502 |
| input2.txt | $O( V ^2 *  E )$ | 0.007666 | $O( V ^3)$     | 0.003318 |

- What are their theoretical run-times (from the previous sections) and how do they compare?  $O(|V|^2 * |E|)$  versus  $O(|V|^3)$
- What are the actual run-times that you computed? Which method is better? Why? The Floyd-Warshall algorithm. It costs less time than the Bellman-Ford algorithm.

## 5 Conclusions

- What did you find difficult about the assignment?  
Adapting the Bellman-Ford algorithm.
- What did you learn?  
I learned about how the google map working and how to revise python code.
- What is one real-world problem that you think each of these problems would be good at solving?  
Find all starbucks within a circle with a given radius, or designing map applications like google map.