CS141 – Intermediate Algorithms and Data Structures
Assignment 2 – All Pairs Shortest Path

Matthew Lauhakamin

November 30, 2016

Abstract

1 Introduction

- What is the problem that you are solving?
  Given a graph with V vertices and E edges, we are finding the path with the cheapest cost.
- What methods are you going to use to solve the problem?
  We are using the Floyd-Warshall algorithm along with the Bellman-Ford algorithm in order to solve the problem.
- Why are these good methods to use?
  These are both popular algorithms with real-world applications. They are also very efficient at solving the problem. They can also both handle negative edge weights effectively.
- Why are you going to be using both of them?
  To compare and contrast the differences in run time and efficiency.

2 Bellman-Ford

- What is the Bellman-Ford algorithm?
  The Bellman-Ford algorithm claims that the longest path in any graph will have at most $V-1$ edges with V being the number of vertices.
- Why are you using it?
  We can use it to find the minimum distance between a vertex against all other vertices.
- How did you adapt it to work for all-pairs as opposed to single source?
  I added an additional loop to iterate through all the vertices.
- What is the run-time of the algorithm before and after your adaptation?
  Before the adaption the run time is $O(|V| * |E|)$. After the adaption the run time becomes $O(|V|^2 * |E|)$.

3 Floyd-Warshall

- What is the Floyd-Warshall algorithm?
  The Floyd-Warshall algorithm is able to find the shortest distance between every pair of vertices.
- Why are you using it?
  To compare it to the Bellman-Ford algorithm.
- How is it better than the Bellman-Ford algorithm? It is designed to solve our problem without adapting.
- What is the run-time of the algorithm?
  The run time is $O(|V|^3)$.

4 Results

- Compare and contrast the two algorithms? What makes one more suited for this problem?

Benchmarks	Bellman-Ford		Floyd-Warshall	
	$O(.)$	Actual	$O(.)$	Actual
input1.txt	$O(\|V\| * \|E\|)$	¡0	$O(\|V\|^3)$	¡0
input10.txt	$O(\|V\| * \|E\|)$	¡0	$O(\|V\|^3)$	¡0
input100.txt	$O(\|V\| * \|E\|)$	2.5	$O(\|V\|^3)$	1.3

The Floyd-Warshall algorithm did not require as many modifications to work. It is faster than the Bellman-Ford algorithm. Bellman-Ford calculates the same thing even if it had been calculated before.

- What are their theoretical run-times (from the previous sections) and how do they compare?
  Bellman-Ford: $O(|V| * |E|)$, Floyd-Warshall: $O(|V|^3)$.
- What are the actual run-times that you computed? Which method is better? Why?
  Bellman-Ford: $O(|V|^4)$, Floyd-Warshall: $O(|V|^3)$.

5 Conclusions

- What did you find difficult about the assignment?
- What did you learn?
- What is one real-world problem that you think each of these problems would be good at solving?