

CS141 – Intermediate Algorithms and Data Structures

Assignment 2 – All Pairs Shortest Path

Xiao Zhou, 861179161

11/30/2016

Abstract

Benchmarks	Bellman-Ford		Floyd-Warshall	
	$O(.)$	Actual	$O(.)$	Actual
input1.txt	$O(n^4)$	0.0002531	$O(n^3)$	0.00009088

1 Introduction

- We are trying to find the shortest path to any vertex from a source
- We will be implementing Bellman-Ford, Floyd-Warshall so solve the shortest path problem.
- The methods can handle negative weight edges which is impossible for Dijkstra's path finding algorithm
- We will use both algorithms to observe their respective runtime.

2 Bellman-Ford

- The Bellman-Ford algorithm is a dynamic programming algorithm that performs edge relaxation on edges of a graph.
- The end result is to find the shortest path within a graph.
- We look at the edge of the vertices as oppose to vertices alone.
- $O(nm)$ before and $O(n^4)$ after

3 Floyd-Warshall

- The Floyd-Warshall algorithm dynamic programming algorithm that performs shortest path on each face of comparison and propagate the edge to the next face, thus finding the shortest path over all.
- We are using it to find the shortest path on a graph
- The algorithm is better than Bellman-Ford's because it has a runtime of $O(n^3)$ as opposed to $O(n^4)$ on all pairs of vertices
- $O(n^3)$

4 Results

- Floyd-Warshall was faster and handles vertice pair better than Bellman-Ford
- $O(n^4)$ for Bellman-Ford, $O(n^3)$ for Floyd-Warshall, therefore Floyd-Warshall is faster
- 0.0002531 for Bellman-Ford, 0.00009088 for Floyd-Warshall, Floyd-Warshall is faster because Bellman-Ford is not really tuned for vertice pairs

5 Conclusions

- The most difficult part of the assignment was trying to figure out how to handle the graph.
- I learned how to analyze and draw conclusion about the pros and cons of Bellman-Ford and Floyd-Warshall algorithms
- We can use these algorithms to find most cost effective way of exchange currency by analyze the rate of exchange between different currencies.