CMPT435 Assignment5

Brian Sprague

December 4th, 2020

1 Introduction

This is my LaTeX document submission for Assignment 5. In this document you will find an explanation of the asymptotic run times for Single Source Shortest Path and Fractional Knapsack.

Figure 1: Java

2 Single Source Shortest Path

2.1 Bellman-Ford

The asymptotic run time of the Bellman-Ford Single Source Shortest Path algorithm is O(VE) where V is equal to the number of vertices and E is equal to the number of edges. This is the worst case rune time, hence the big O notation. The reason that it comes out to O(VE) is because the run time of the algorithm

is dependant on the number of both of these values. The formula is also just a simple multiplication of variables, nothing fancy like log or squared.

2.2 Dijkstra's Algorithm

Dijkstra's Algorithm is an alternate way of finding the single source shortest path. It can be used as an alternative to the Bellman-Ford algorithm, with the difference being that it can not account for negative weights, which is where the Bellman-Ford algorithm gets its use. Dijkstra's Algorithm has a faster asymptotic run time than the Bellman-Ford Algorithm. It's run time for an undirected graph is O(V2). There are other implementations of Dijkstra's Algorithm, and they sometimes have different run times. These include: Dijkstra's Algorithm with list at O(V2), Dijkstra's Algorithm with binary heap at O((E+V)logV), and Dijkstra's Algorithm with Fibonacci heap at O(E+VlogV).

3 Fractional Knapsack

The asymptotic run time of the fractional knapsack algorithm is O(nlogn). This is because it involves a sort for all of your items, so if you for example to merge sort, it will take O(nlogn). Then, you would iterate through your items in order decreasing ratio, and on each iteration it will take O(n). So, this makes the total complexity O(nlogn * n) = O(nlogn).