Vince Bjazevic

Robert Sapien

Project 3 Report

A close up of a map

Description automatically generated

Based on the results of the test files we can see that the Non-Heap Greedy uses merge sort to solve the problem making the Non-Heap algorithm’s running time come out to O(nlogn). On the contrary we can see that the heap based method has a running time of O(n + klogn) this is due to the sorting and sifting down associated with heap structures.

Graph for 1a/1b

A screenshot of a cell phone

Description automatically generated

These graphs a put side by side on the graph to show growth rates of time for Traditional vs Space-Efficient. Traditional programing approach shows a faster time over the Space-Efficient approach sacrificing speed for reduced space. Space is not included in this model just time in reference to file number.

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Graph 1a plots the line of the Traditional Dynamic Programming Knapsack problem with a time space comparison. As capacity by n increases so does the time, with an efficiency of O(nw). Graph 1b plots the line of the Space-Efficient Dynamic Programming Knapsack problem. Like 1a, with the capacity increasing, the time to compute the algorithm increases as well. Comparing the two, algorithm 1b has a more significant time-space trade-off than 1a. The less space you utilize, the more time you need to spend to calculate the algorithm.