Project 2 Report

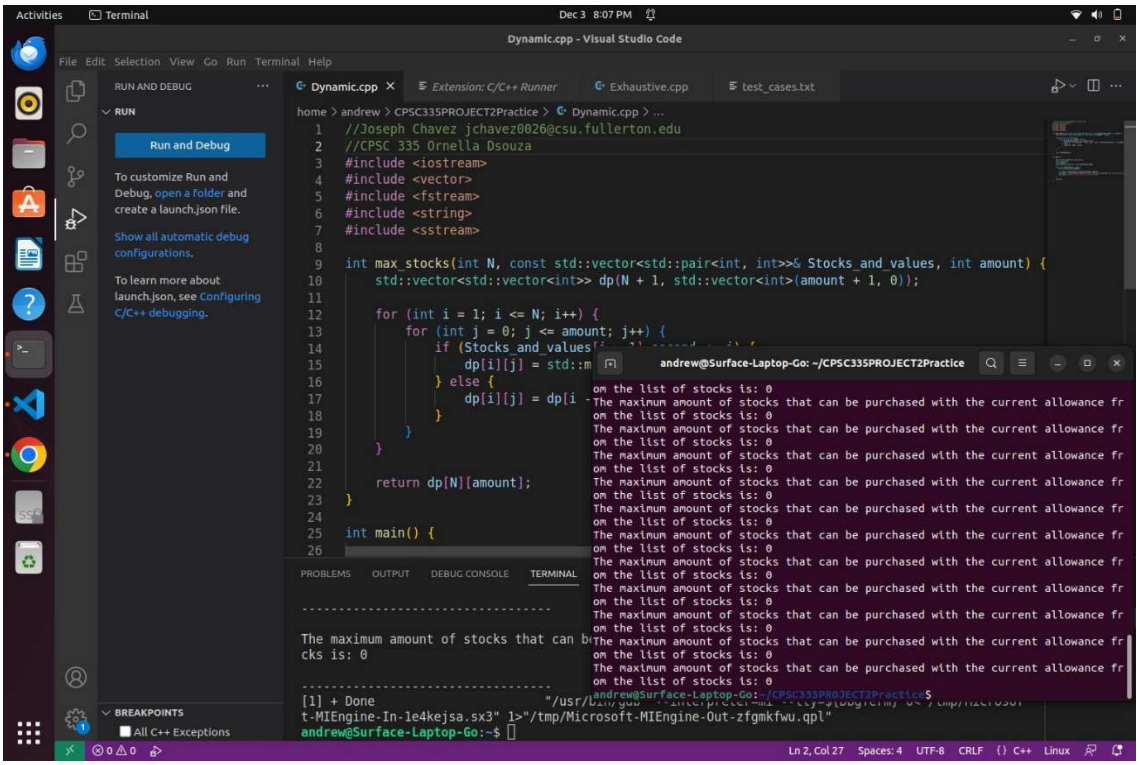
Group Member

Joseph Chavez jchavez0026@csu.fullerton.edu

This is my submission for Project 2 Exhaustive vs Dynamic

https://github.com/JosephChavxz/CPSC335-Project-2

Tuffix Screenshot:



Part A Exhaustive: Time complexity

The time complexity of the code can be approximated as O(n * 2^n).

Part A Exhaustive: Logic

The overall approach is exhaustive, examining each possible combination of stock quantities to find the optimal purchase under the budget constraint. This method can be computationally expensive due to its brute-force nature, especially for large datasets.

Part B Dynamic: Time complexity

The time complexity of the program is O(N * amount).

Part B Dynamic: Logic

The code aims to efficiently compute the optimal number of stock units to purchase within a budget, considering the price and availability of each stock type.

Which approach is better?

Between the two approaches exhaustive and dynamic programming, the dynamic programming approach is generally better for stock maximization. Efficiency: Dynamic programming is more efficient in terms of time complexity. The brute-force approach has an exponential time complexity due to generating all combinations of stocks, while dynamic programming has a polynomial time complexity, specifically O(N * amount). Scalability: The dynamic programming approach scales better with larger inputs. The brute-force method becomes impractical as the size of the input increases, as it needs to evaluate an exponentially growing number of combinations. Optimization: Dynamic programming is designed to solve optimization problems like this one, where the objective is to find the best solution (maximum stocks) under certain constraints (budget). The dynamic programming approach is preferable for larger datasets and for efficiently finding the optimal solution in the stock maximization problem.