Assignment 4 Part 2

Re-write your program from part one of this assignment using a dynamic programming approach. In your code, write comments giving the time-complexity of your previous approach (whatever it may be) compared with the time complexity of a dynamic programming approach.

You may have already implemented a dynamic programming approach in part 1. That's good luck for you! You can turn in the same thing again. BUT, you will need to be able to identify that the code you wrote does indeed qualify as "dynamic programming." I will leave that analysis up to you.

Put comments in your code that explain the dynamic programming (table based) approach.

Pro-Tip: Sort the input list first.

You should notice that `main.cpp` in your repo has been modified for part 2. It now contains several test cases and a "random number" test at the end. The random number test creates a massive 28,000 value input vector. Solving the problem with such a large input is not feasible using recursion, but with dynamic programming it should take just a few seconds. My dynamic programming solution code, running on my computer, takes less than 10 seconds to solve it!

Submission: Your program will be submitted using GitHub classroom.

1) Use the existing GitHub classroom repo you started in part 1 of this assignment.

2) Continue to edit and write code in the repo on your computer. Use git add, git commit, and git push to put that code onto github. I will have access to the repo. You can make as many commits as you like for this assignment with no penalty.

3) I will grade the final commit made before the due-date.

4) You don't have to make any submission on canvas.