CS3000: Algorithms & Data — Fall '18 — Jonathan Ullman

Homework 4
Due Friday October 12 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday October 12 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. *Strategery*

Alice and Bob play the following game. There is a row of n tiles with values $a_1, \ldots, a_n$ written on them. Starting with Alice, Alice and Bob take turns removing either the first or last tile in the row and placing it in their pile until there are no tiles remaining. For example, if Alice takes tile 1, Bob can take either tile 2 or tile n on the next turn. At the end of the game, each player receives a number of points equal to the sum of the values of their tiles minus that of the other player's tiles. Specifically, if Alice takes tiles $A \subseteq \{1, \ldots, n\}$ and Bob takes tiles $B = \{1, \ldots, n\} \setminus A$, then their scores are

$$\sum_{i \in A} a_i - \sum_{i \in B} a_i \quad \text{and} \quad \sum_{i \in B} a_i - \sum_{i \in A} a_i,$$

respectively. For example, if $n = 3$ and the tiles have numbers $10, 2, 8$ then taking the first tile guarantees Alice a score of at least $10 + 2 - 8 = 4$, whereas taking the last tile would only guarantee Alice a score of at least $8 + 2 - 10 = 0$.

In this question, you will design an algorithm to determine the maximum score that Alice can guarantee for herself, assuming Bob plays optimally to maximize his score. Note that the sum of their scores is always 0, so if Bob is playing optimally to maximize his own score, then he is also playing optimally to minimize Alice's score.

(a) Describe the set of subproblems that your dynamic programming algorithm will consider. Your solution should look something like "For every ..., we define OPT(...) to be ..."

Solution:

(b) Give a recurrence expressing the solution to each subproblem in terms of the solution to smaller subproblems.

Solution:

(c) Explain in English a valid order to fill your dynamic programming table in a "bottom-up" implementation of the recurrence.

Solution:

(d) Describe in pseudocode an algorithm that finds the maximum score that Alice can guarantee for herself. Your implementation may be either "bottom-up" or "top-down."

Solution:

(e) Analyze the running time and space usage of your algorithm.

Solution:

Problem 2. *Gerrymandering*

Gerrymandering is the practice of constructing electoral districts to lead to outcomes that favor a particular party. In this problem we will explore an algorithmic question that arises in the act of gerrymandering.

Suppose there are n *precincts* $P_1, \ldots, P_n$, where n is even, each containing M registered voters. We must divide these into two *districts*, each consisting of exactly $n/2$ precincts. For each precinct P_i, our research has shown that there are $V_{i,W}$ voters for the Whig party and $V_{i,B}$ voters for the Bull Moose Party where $M = V_{i,W} + V_{i,B}$. Our goal is to decide whether or not it is possible to gerrymander the districts so that the Whig party gets at least $\frac{Mn}{4} + 1$ votes (i.e. a majority) in each of the two districts.

For example, if we are given the following values with $n = 4, M = 100$,

Precinct	P_1	P_2	P_3	P_4
Votes for Whig Party	55	43	60	47
Votes for Bull Moose Party	45	57	40	53

then we can make one district consisting of $\{P_1, P_4\}$ and the other district consisting of $\{P_2, P_3\}$ so that the Whig party gets 102 votes in the first district and 103 votes in the second district. Thus the answer would be YES. That is, for this problem you do not need to consider how to find such a set of districts.

(a) Describe the set of subproblems that your dynamic programming algorithm will consider. Your solution should look something like "For every [...], we define S([...]) to be [...]." Note that here your algorithm only needs to output a Boolean answer YES or NO.

Solution:

(b) Give a recurrence expressing the solution to each subproblem in terms of the solution to smaller subproblems.

Solution:

(c) Explain in English a valid order to fill your dynamic programming table in a "bottom-up" implementation of the recurrence.

Solution:

(d) Describe in pseudocode an algorithm that determines if it is possible to gerrymander the two districts so that the Whig party holds a majority of the votes in both districts. Your implementation may be either "bottom-up" or "top-down."

Solution:

(e) Analyze the running time and space usage of your algorithm.

Solution: