CS41 Homework 9

This homework is due at 11:59PM on Sunday, November 11. Write your solution using LaTeX. Submit this homework using **github** as a **.tex** file. This is a **partnered homework**. You should primarily be discussing problems with your homework partner.

It's ok to discuss approaches at a high level with others. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner *while in lab*. In this case, note (in your **README** file) who you've worked with and what parts were solved during lab.

1. **Implementing RNA Substructure.** For this problem, your task is to write a program that takes as input the name of a file containing a single string representing and RNA molecule and outputs the size of the largest matching according to the rules for RNA Substructure discussed in class.

   (a) You may write your program in C++ or Python.

   (b) You're welcome to test your program using test files in `/home/brody/public/cs41/rna_test_data`.

   (c) You're welcome to test your program answers against the rna implementations `rna-A` and `rna-B`, located in `/home/brody/public/cs41`.

   (d) The runtime of your algorithm should be comparable to the more efficient of `rna-A` and `rna-B`.

2. **Cassie's Convenience Stores.** Cassie plans to open a chain of convenience stores along Baltimore Pike. Using market research, Cassie identified a series of n locations where she can open stores. For each location, Cassie calculated (again using market research) how much annual profit she is likely to gain by placing a store at this location. She can build as many convenience stores as she wants, as long as they are not too close (otherwise, they will compete with each other for business and lose money).

   In this problem, you will design an algorithm that helps Cassie determine how much annual profit she can make. The input to this problem is an integer K, and two arrays $L[1 \cdots n]$ and $P[1 \cdots n]$. Assume that $0 \leq L[i] \leq N$ for each i[1], and that L is sorted in increasing order. The goal of this problem is to output the maximum possilbe profit by placing convencience stores at locations from $L[1 \cdots n]$ such that the distance between any two locations is at least K.

   (a) If Cassie decides to build a convenience store at location $L[k]$, what is the closest location to the east or west that she can build, given her list of locations? Write an algorithm $West[k]$ that returns the index of the closest location k' to k such that $L[k'] < L[k]$. Write a similar algorithm for $East[k]$.

   (b) Design a dynamic program that computes the maximum annual profit Cassie can earn by placing her convenience stores.

   (c) Modify your dynamic program so it returns the set of locations that maximize Cassie's profit.

3. **Gerrymandering (K& T 6.24)** *Gerrymandering* is the practice of carving up electoral districts in very careful ways so as to lead to outcomes that favor a particular political party. Recent court challenges to the practice have argued that through this calculated redistricting, large numbers of voters are being effectively (and *intentionally*) disenfranchised.

   Suppose we have a set of n precincts $P_1, \ldots, P_n$, each containing m registered voters. We're supposed to group these precincts into two *districts*, each consisting of $n/2$ precincts. Now, for each precinct, we have information on how many voters are registered to each of two political parties. Say that the

[1] $L[i]$ represents the the location of store i, where 0 is the westernmost terminus of Baltimore Pike, and N is the easternmost terminus

set of precincts is *susceptible* to gerrymandering if it is possible to set up two districts in such a way that the same party holds a majority in both districts.

Give an algorithm to determine whether a given set of precincts is susceptible to gerrymandering. The running time of your algorithm should be polynomial in n and m.

For example, suppose there are four precincts, and two political parties A and B. Letting A_i and B_i be the number of voters in precinct i of each political party, Suppose we have

$$A_1 = 55, A_2 = 43, A_3 = 60, A_4 = 47 \textit{ and}$$

$$B_1 = 45, B_2 = 57, B_3 = 40, B_4 = 53\ .$$

This set of precincts is susceptible to gerrymandering since pairing precincts 1 and 4 together and $2, 3$ together gives party A a 102 - 98 majority in the first district and a 103 - 97 majority in the second.

Hint: Focus on the choice you need to make as you're building up a partial solution to this problem (in this case, an assignment of precincts to districts). You will likely need to maintain extra information about the partial solution.

4. **(extra challenge) Cassie's Convencience Stores Redux.** It is natural to assume that the profit of a convenience store changes depending on how close to other convience stores it is. Suppose instead of an array of profits $P[1 \cdots n]$ and a minimum distance between stores of K, Cassie does more market research and gets information on $P[k, d]$, the annual profit of location k assuming that the closest other store is at d km away.

   Design an algorithm that computes Cassies maximum profit.