CS41 Homework 8

This homework is due at 11:59PM on Monday, November 1. Write your solution using LaTeX. Submit this homework using **github** as a **.tex** file; the code should be in a file called either **rnasubstructure.cpp** or **rnasubstructure.py**. 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 teammate *while in lab*. In this case, note (in your **homework submission poll**) who you've worked with and what parts were solved during lab.

1. (K&T 5.1) You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains n numerical values (so there are $2n$ values total). You'd like to determine the *median* of this set of $2n$ values, defined as the n-th smallest value. (**Note** that the median *must* be an element in the list; this might be different than previous definitions of "median" that you have encountered.)

   The only way you can access these values is through *queries* to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the k-th smallest value it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.

   - Design an algorithm that finds the median value using at most $O(\log n)$ queries. Full pseudocode is not necessary, but you must clearly explain how it works, and you must handle all edge cases; e.g., do not assume that n is even.
   - Show that your algorithm correctly returns the median.
   - Prove that your algorithm uses only $O(\log n)$ queries.

2. **Implementing RNA Substructure Dynamic Program**

   For this problem, your task is to write a dynamic program (an actual *program*!) that takes as input an RNA string and outputs the size of the largest matching, according to the rules of the RNA Substructure problem described in class.

   - You may write your program in C++ or in Python.
   - If you write in C++, we encourage you to use the limited starting code we've provided. The program is invoked by running `./rna <name of test file>`. If you write in python, the program should be invoked by running `python rna.py <name of test file>`.
   - Your solution must be a dynamic program.
   - Your solution must be efficient—graders will test your program against large RNA substrings, and there will be nontrivial penalties for programs that take too long.
   - The file format consists of a single string of letters from $\{A, C, G, U\}$. (There are some short examples in `/home/fontes/public/cs41/rna-test-data/`.) Your program should read in the file, compute the largest RNA substructure size, and output the result. For example:

```
> ./rna /home/fontes/public/cs41/rna-test-data/test1
The largest RNA matching is: [2]
```

3. **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 convenience 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.

4. **(extra challenge) Cassie's Convenience Stores (v2.0)** It is natural to assume the profit of a convenience store changes depending on how close to other convenience stores it is. Suppose instead of an array of profits $P[1 \cdots n]$ and a minimum distance between stores 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 least d km away.

   Design an algorithm that computes Cassie's maximum profit.

[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