Problem Set 3

This problem set is due at **11:59 pm** on **Wednesday, September 30th, 2015**. The exercises are **optional**, and should not be submitted.

Exercise 2- 1: Exercise 30.1-7 in CLRS

Exercise 2- 2: Exercise 30.2-5 in CLRS

Exercise 2- 3: Exercise 30.2-7 in CLRS

Exercise 2- 4: Exercise 23.1-5 in CLRS

Exercise 2- 5: Exercise 23.1-9 in CLRS

Problem 3- 1: Key Word Search

Alice and Bob are working together to patent a new algorithm called the SDFFFT (Super Duper Fast Fast Fourier Transform). One day, Alice decides to send her write up of the patent over to Bob as a message M, a binary string of length r.

Competitor Eve is determined to discover the secrets behind the algorithm, and intercepts the message as it is being sent. Eve suspects there are two possible improvements Alice could have made to the FFT algorithm in her patent. In order to identify the exact improvement, Eve guesses two key words, S_1 and S_2, representing the first and second improvement, respectively, and wants to check which word appears more frequently in the message sent by Alice. Both S_1 and S_2 are binary strings of length s, where $s \leq r$. Note here that a keyword can match multiple, possibly overlapping, substrings in the intercepted message as long as each begins at a different position.

Unfortunately, the line that Eve uses to intercept the message is noisy, so some of the bits in the message she intercepts may be incorrect. For example, if Alice transmits message 100100, Eve may obtain a message that resembles 110101, where the second and last bits are wrong due to channel noise. To account for this noise problem, Eve considers there to be match between a keyword and the intercepted message at location i if there are less than or equal to e errors in the match between the key word and the substring which begins at location i. Going back to the example, if one of Eve's guesses is 100 and $e = 1$, then the number of matches to that guess would be 3, because 100 matches 110 (substring starting at position 0), 101 (substring starting at position 1), and 101 (substring starting at position 3).

Your objective is to help Eve count the number of times her two guesses S_1 and S_2 appear in the message she intercepts, and find out which one appears more frequently.

Put another way, we have three binary strings M, S_1, S_2, of lengths r, s, s, respectively, where $s \leq r$. We want to count the number of times S_1 and S_2 appear as a substring of M with $\leq e$ bit errors, and return the substring (S_1 or S_2) that appears more frequently in M.

(a) Eve's first attempt at solving this problem is to use a brute force algorithm that runs in $O(rs)$. Describe an algorithm that achieves this run time and show why it is correct.

(b) After implementing her brute force algorithm, Eve is frustrated by how slow the program runs. She is certain there is a faster way of solving this problem using FFTs. Help her construct polynomials $P(x)$, $Q_1(x), Q_2(x)$ of degrees $r-1, s-1, s-1$ respectively such that $P(x)Q_1(x)$ (respectively $P(x)Q_2(x)$) contain coefficients that track the number of bits that differ between S_1 (respectively S_2) and the length s substring of M beginning at each possible position. Explain how your choice of $P(x)$, $Q_1(x), Q_2(x)$ maintains this property.

Hint: Consider choosing coefficients for P, Q_1, Q_2 that somehow match up with the bits in M, S_1, S_2 respectively. Keep in mind that you don't need to worry about the value of the polynomials for any particular value of x, as we only care about their coefficients.

(c) Using the polynomials you helped Eve construct in part (b), formulate an $O(r\log(r))$ algorithm that solves Eve's guess counting problem.

Problem 3- 2: Optic Fiber Network

Entrepreneur Ben Bitdiddle wants to expand his optic fiber company into the country of Hackerland. His goal is to build an optical fiber network that connects all the cities in Hackerland with the least amount of material. Knowing that you have taken 6.046, Ben decides to hire you, an MIT engineer, to help him to decide where to place the fibers.

The first idea that came to your mind was the Minimum Spanning Tree T of the cities, when considered as a graph $G = (E, V)$ where V denotes the set of cities and E denotes how the cities are connected. However, to your dismay, Alyssa P. Hacker, the owner of another optical fiber company, has already laid her cables in the configuration of the *unique* MST T of G. Since governmental regulations forbid building an exact same network, you have to come up with another way to connect all the cities while keeping costs low.

Ben asks you if it is possible to build a network in the configuration of a spanning tree with the second least cost, one that is just barely above the cost of building T.

(a) Let G be the weighted undirected graph of the cities, and T be its *unique* minimum spanning tree. Prove that a spanning tree with the second least total length can be obtained by replacing exactly one edge of T.

Hint: Consider a spanning tree S with the second least total length, and that S differs from T at more than 1 edge. Then consider the cycles in $T \cup S$; what can we do?

(b) Given G and T, compute the 2-dimensional array D such that for all pairs of vertices u and v, $D[u, v]$ gives the longest edge on the unique path between u and v in T.

(c) Given T, and using the results from (a) and (b), find a spanning tree with the second least total length in $O(V^2)$ time.