CS 170 Homework 2

Due **9/13/2021, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

In addition, we would like to share correct student solutions that are well-written with the class after each homework. Are you okay with your correct solutions being used for this purpose? Answer "Yes", "Yes but anonymously", or "No"

2 Werewolves

You are playing a party game with n other friends, who play either as werewolves or citizens. You do not know who is a citizen and who is a werewolf, but all your friends do. There are always more citizens than there are werewolves.

Your goal is to identify one player who is certain to be a citizen.

Your allowed 'query' operation is as follows: you pick two people. You ask each person if their partner is a citizen or a werewolf. When you do this, a citizen must tell the truth about the identity of their partner, but a werewolf doesn't have to (they may lie or tell the truth about their partner).

Your algorithm should work regardless of the behavior of the werewolves.

(a) Give a way to test if a single player is a citizen using $O(n)$ queries. Just an informal description of your test and a brief explanation of why it works is needed.

(b) Show how to find a citizen in $O(n \log n)$ queries (where one query is taking two people x and y and asking x to identify y and y to identify x).

There is a linear-time algorithm for this problem, but you cannot use it here, as we would like you to get practice with divide and conquer.

Hint: Split the group into two groups, and use part (a). What invariant must hold for at least one of the two groups?

Give a 3-part solution.

(c) (**Extra Credit**) Can you give a linear-time algorithm?

Hint: Don't be afraid to sometimes 'throw away' a pair of people once you've asked them to identify their partners.

3 Fourier Transform Basics

We will use ω_n to denote the first root of unity $\omega_n = e^{2\pi i/n}$.

Fast Fourier Transform! The *Fast Fourier Transform* $\text{FFT}(p, n)$ is an algorithm to perform the *Discrete Fourier Transform* $\text{FT}(p)$ which takes arguments n, some power of 2, and p, some vector $[p_0, p_1, \ldots, p_{n-1}]$.

Treating p as a polynomial $P(x) = p_0 + p_1 x + \ldots + p_{n-1}x^{n-1}$, the FFT computes the following matrix multiplication in $\mathcal{O}(n \log n)$ time:

$$\begin{bmatrix} P(1) \\ P(\omega_n) \\ P(\omega_n^2) \\ \vdots \\ P(\omega_n^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n^1 & \omega_n^2 & \cdots & \omega_n^{(n-1)} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{(n-1)} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{bmatrix}$$

(a) What is the Fourier transform of $(3, i, 2, 4)$?

(b) Find x and y such that $\text{FT}(x) = (5, 2, 1, -i)$ and $\text{FT}(y) = (4, 4, i, i)$.

(c) *Using part b*, find z such that $\text{FT}(z) = (1, -2, 1 - i, -2i)$. (*Hint:* Observe that $\text{FT}(v)$ is a linear transform of v, and recall the properties of linear transforms).

(d) Compute $(2x^2 + 1)(x + 4)$ using a Fourier transform. (*Hint:* Recall that to multiply two polynomials, first, they must both be converted by the Fourier transformation, then multiplied pointwise, and finally converted back to coefficient form)

4 Modular Fourier Transform

Fourier transforms (FT) have to deal with computations involving irrational numbers which can be tricky to implement in practice. Motivated by this, in this problem you will demonstrate how to do a Fourier transform in modular arithmetic, using modulo 5 as an example. This problem is just about understandin the Fourier transform itself; no need to use the FFT algorithm.

(a) There exists $\omega \in \{0, 1, 2, 3, 4\}$ such that ω are 4^{th} roots of unity (modulo 5), i.e., solutions to $z^4 = 1$. When doing the FT in modulo 5, this ω will serve a similar role to the primitive root of unity in our standard FT. Show that $\{1, 2, 3, 4\}$ are the 4^{th} roots of unity (modulo 5). Also show that $1 + \omega + \omega^2 + \omega^3 = 0 \pmod 5$ for $\omega = 2$.

(b) Using the matrix form of the FT, produce the transform of the sequence $(1, 0, 0, 3)$ modulo 5; that is, multiply this vector by the matrix $M_4(\omega)$, for the value $\omega = 2$. Be sure to explicitly write out the FT matrix you will be using (with specific values, not just powers of ω). In the matrix multiplication, all calculations should be performed modulo 5.

(c) Write down the matrix necessary to perform the inverse FT. Show that multiplying by this matrix returns the original sequence. (Again all arithmetic should be performed modulo 5.)

(d) Now show how to multiply the polynomials $2x^2 + 3$ and $-x + 3$ using the FT modulo 5.

5 Counting k-inversions

A *k-inversion* in a bitstring b is when a 1 in the bitstring appears k indices before a 0; that is, when $b_i = 1$ and $b_{i+k} = 0$, for some i. For example, the string 010010 has two 1-inversions (starting at the second and fifth bits), one 2-inversion (starting at the second bit), and one 4-inversion (starting at the second bit).

Devise an algorithm which, given a bitstring b of length n, counts all the k-inversions, for each k from 1 to $n - 1$. Your algorithm should run faster than $\Theta(n^2)$ time. You can assume arithmetic on real numbers can be done in constant time.

Give a 3-part solution.