Carnegie Mellon University Instructor: Venkatesan Guruswami
15-252: More GITCS Term: Spring 2020

Problem Set 5

Due date: Friday, February 20

Instructions

- You are allowed to collaborate with one other student taking the class, or do it solo.
- Collaboration is defined as discussion of the lecture material and solution approaches to the problems. Please note that *you are not allowed to share any written material and you must write up solutions on your own.* You must clearly acknowledge your collaborator in the write-up of your solutions.
- Solutions must be typeset in LaTeX and emailed to the TA.
- You should not search for solutions on the web. More generally, you should try and solve the problems without consulting any reference material other than what we cover in class and any provided notes.
- Please start working on the problem set early. Though it is short, the problem(s) might take some time to solve.

1. Let S be a subset of $\{1, 2, \ldots, n\}$. Give an algorithm to compute the number of 3-term arithmetic progressions in S in time $O(n \log n)$.

   A k-term arithmetic progression is a set of elements $\{a, a+d, a+2d, \ldots, a+(k-1)d\}$, where a and d are integers and $d > 0$. For example, in the set $\{1, 2, 3, 5\}$, there are two 3-term arithmetic progressions:$\{1, 2, 3\}$ and $\{1, 3, 5\}$.

   (Hint: Try to think of polynomials representing the set.)

2. In this exercise, we consider a transform similar to the Discrete Fourier Transform done in class. Let n be a positive integer, and let $N = 2^n$. We define the $N \times N$ matrix DWT_N as follows- index the rows and columns by n length bit vectors, and $\mathrm{DWT}_N(x, y) = (-1)^{x.y}$, where $x.y$ is the dot product of the two vectors x, y i.e. $\sum_{i=1}^{n} x_i y_i (\mathrm{mod}\ 2)$. Describe an algorithm running in time $O(N \log N)$, that, given a vector $a \in \mathbb{R}^N$, computes $\mathrm{DWT}_N a$.

   (Hint: Try to recurse by solving two instances with $N' \times N'$ matrices where $N' = 2^{n-1}$.)