CECS 528, Homework Assignment 2, Spring 2025, Dr. Ebert

Directions: Please review the `Homework` section on page 6 of the syllabus including a list of all rules and guidelines for writing and submitting solutions.

Due Date: Monday, February 10th as a PDF-file upload to the HW2 Canvas dropbox.

Problems

1. Apply both the FFT and IFFT algorithms for determining the coefficients of the product $C(x)$ of the linear polynomials $A(x) = 2x - 5$ and $B(x) = -3x + 4$. Do so as follows.

   (a) Compute $\text{DFT}_4(A)$ using FFT. (5 points)

   (b) Compute $\text{DFT}_4(B)$ using FFT. (5 points)

   (c) Compute $\vec{c} = \text{DFT}_4(C)$ by multiplying corresponding components of $\text{DFT}_4(A)$ and $\text{DFT}_4(B)$. (5 points)

   (d) Compute $\text{DFT}_4^{-1}(\vec{c})$ using IFFT. (5 points)

   (e) Verify that $\text{DFT}_4^{-1}(\vec{c})$ gives the correct coefficients of $A(x) \cdot B(x)$. (5 points)

   (f) Why do we need to work with the fourth roots of unity? What roots of unity would we work with if A and B were both quadratic polynomials? Explain. (5 pts)

   Note: please show the calculation for each subproblem of each of the three recursion trees (two FFT trees and one IFFT tree). Points will be lost otherwise.

2. Suppose that a CPU of some machine M can, on average, process a single machine instruction in 5×10^{-9} seconds. Moreover, an instance of some problem is an $n \times n$ matrix R, and there are four known algorithms ($\mathcal{A}_1, \ldots, \mathcal{A}_4$) for solving a given problem instance. When these four algorithms are programmed and compiled on M, the average number of instructions (as a function of the number n of rows and columns of matrix R) that need to be executed by each program is i) $1500n$, ii) $250n \log n$, iii) $45n^2$, and iv) $6n^3$, respectively. What is the largest input matrix that each program can process and solve within a one-hour period? Show all work for each calculation except to ii) which you may solve with the help of software. (20 points)

3. An instance of problem `L` is an array a of n natural numbers and an integer $t \geq n$. Describe an effficient algorithm for finding the least integer d for which

$$\sum_{i=0}^{n-1} \left\lceil \frac{a[i]}{d} \right\rceil \leq t.$$

   First describe your algorithm in one or more paragraphs, followed by providing pseudocode. Finish by analyzing the number of steps your algorithm takes as a function of both n and $\log t$, the two size parameters for `L`. Assume that each arithmetic operation requires O(1) steps. (25 points)