Homework 3

Start Assignment

Due Feb 5 by 8am **Points** 20 **Submitting** a file upload **File Types** pdf
Available Jan 29 at 8am - Feb 5 at 8am

Suggested reading

Chapter 2 [DPV].

Practice Problems (do not turn in)

[DPV] Problem 2.7 (Sum and product of roots of unity**)**

[DPV] Problems 2.8, 2.9(a) (FFT practice)

(FFT design)

Let $A(x) = 1 - 6x - 2x^2 + 7x^3$. You wish to run FFT to evaluate this polynomial.

(a) What is $A_{odd}(y)$ and $A_{even}(y)$?

(b) What is the appropriate root of unity to use?

(FFT as a black box)

Design an algorithm that takes as input a set $S = \{s_1, s_2, \ldots, s_n\}$ of distinct natural numbers such that $0 \leq s_i \leq 100n$, and a natural number N, and outputs True if the equation $s_i + s_j + s_k = N$ has at least one solution, and return False otherwise. There is a simple solution that runs in $O(n^3)$ time (can you find an $O(n^2)$ solution?), but you can improve on these times using FFT! Your faster algorithm should use FFT as a black box. That is, describe how to form the inputs to FFT and what can be done with the output from FFT to solve the problem.

Example: For N = 6 and S = { 1; 2; 3; 5; 10 } your design should output True since 1+2+3 = 6. For N = 20 and the same set S the answer should be True again since 5+5+10 = 20 (yes, you can have $s_i = s_j = s_k$) but for N = 19 the answer is False since no three numbers add up to 19.

Graded Problem

A thief is planning on burglarizing some subset of n consecutive houses in a neighborhood. The houses are labeled 1,2,..., n and the thief will address then sequentially. The thief has an estimate of the profit to be earned from burglarizing each house $p_i, i = 1 \ldots n$, where $p_i > 0$. To avoid detection, he decides that he will never burglarize two adjacent houses, meaning that if he burglarize house 2, he cannot

burglarize house 1 or house 3. Design a dynamic programming algorithm to determine the maximum total profit he can achieve.

Example: In each of the following two neighborhoods, the maximum achievable profit is 100:

Case 1: p = [20, 100, 30].

Case 2: p = [40, 30, 10, 60].

Your input is the list $[p_1, p_2, \ldots, p_n]$. Your output should be the maximum profit the thief can get. You don't have to return the list of houses the thief has to burglarize to achieve the maximum.

Please answer the following parts:

1. Define the entries of your table in words. E.g. T(i) or T(i, j) is ...
2. State a recurrence for the entries of your table in terms of smaller subproblems. Don't forget your base case(s).
3. Write pseudocode for your algorithm to solve this problem.
4. State and analyze the running time of your algorithm.

Faster (in asymptotic Big O notation) and correct solutions are worth more credit.

You will upload a pdf of your typed solution. **Handwritten solutions will be penalized.** Please see the related threads in Ed Discussions for detailed expectations regarding your submission.