E0 225 : Homework 2

Deadline : 12th September, 2023, 2 pm

Instructions

- All problems carry equal weight.
- You need to attempt the homework problems on your own (no peer discussions are allowed). You are also forbidden from consulting the internet and other external sources.
- Academic dishonesty/plagiarism will be dealt with severe punishment.
- Late submissions are accepted only with prior approval or medical certificate.

1. *D&C: Balanced partition.*
   Let G be any binary tree with n vertices with n to be an even number.
   - Show that by removing a single edge, we can partition the vertices of G into two sets $\mathcal{X}$ and $\mathcal{Y}$ such that there are no edges between the sets $\mathcal{X}$ and $\mathcal{Y}$ and $n/4 \le |\mathcal{X}| \le |\mathcal{Y}| \le 3n/4$.
   - Give an example of a simple binary tree whose most evenly balanced partition upon removal of a single edge has $|\mathcal{Y}| = 3n/4$.
   - Show that for any $k = 1, 2, \ldots, n-1$, by removing at most $3\log_2(n)$ edges, we can partition the vertices of G into two sets $\mathcal{X}, \mathcal{Y}$ such that there are no edges between the sets $\mathcal{X}$ and $\mathcal{Y}$ and $|\mathcal{X}| = k$ and $|\mathcal{Y}| = n - k$.

2. *FFT for setSum.*
   Let S and T be two sets of integers in the range $[0, z-1]$ where z is a power of two. Compute the following in $O(z \log_2 z)$ time:

   (a) all elements contained in the set $S + T := \{s + t : s \in S, t \in T\}$,

   (b) for each element $u \in S + T$, the number $k_u := |\{(s,t) \in S \times T : s + t = u\}|$.

   Hint: Find some polynomials p_S, p_T of degree $< z$ that represents the sets S and T.

3. *D&C: Median.*
   We are given two sets A and B containing integers and both sets are stored as (increasingly) ordered arrays. Our goal is to compute the median of their disjoint union $A \amalg B$. Show how to do that in time $O(\log(|A| + |B|))$ using a divide-and-conquer based algorithm.

4. *Dynamic Programming: Optimizing space in a library.*
   A library bought a new bookcase. The shelves all have the same fixed width $S > 0$ but the vertical distances between shelves can be adjusted individually according to the heights of the books placed on them. The order in which the n books are placed on the shelves is fixed by the cataloging system of the library: $b_1, b_2, \ldots, b_n$. A book b_i has width $w_i > 0$ and height $h_i > 0$. Using dynamic programming, design an algorithm that places the books on the shelves such that the total space usage, i.e. the sum of the heights of the largest book on each shelf, is minimized. Argue that your algorithm always finds an optimal solution and state its time complexity.

5. *Dynamic programming: Largest monochromatic square.*
An axis-aligned square Q with side length $n \in \mathbb{N}$ is subdivided into n^2 squares of side length 1 that are all parallel to the axes. Each one of the small squares is either colored grey or white. The goal is to determine the side length and the position of the biggest square in Q that is completely colored in white and parallel to the axes. In case there is more than one such square, any one of those may be chosen. See Figure 1 for an example. Design an algorithm that solves the problem and employs dynamic programming. Describe your algorithm and argue that it is correct. Include its complete recurrence relation.

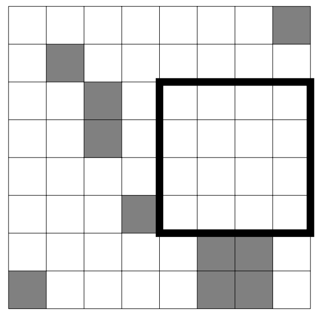

Figure 1: The bold edging marks a biggest square that is completely colored in white.

Recommended practice problems: (not for submission)
Divide and Conquer:
– Kleinberg-Tardos: Exercises 2, 3, 5, 6, 7 from Chapter 5.
– CLRS (3rd ed): 30.1-4, 30.1-5, 30.2-2, 30.2-7.
Dynamic Programming:
– CLRS (3rd ed): 15.3-6, 15.4-6, 15.1, 15.2, 15.3, 15.4, 15.5, 15.7.
– Erickson Chapter 3 [1] . Problems 4, 5, 6, 7, 8, 9, 10.

[1] `https://jeffe.cs.illinois.edu/teaching/algorithms/book/03-dynprog.pdf`