CS331: Algorithms and Complexity
Homework I

Trung Dang Ryan Park Kevin Tian

Due date: September 11, 2024, end of day (11:59 PM), uploaded to Canvas.

Late policy: 15% off if submitted late, and 15% off for every further 24 hours before submission.

Please list all collaborators on the first page of your solutions.

When runtimes are unspecified, slower runtimes than the intended solution receive partial credit.

1 Problem 1

Let S and W be two Array instances both containing n positive real numbers, and denote their i^{th} entries by $S[i]$ and $W[i]$ respectively for all $i \in [n]$. Each $W[i]$ denotes a weight assigned to $S[i]$. Suppose all numbers in S are distinct, and for each $i \in [n]$, let

$$\ell_i := \sum_{\substack{j \in [n] \\ S[j] < S[i]}} W[j], \; g_i := \sum_{\substack{j \in [n] \\ S[j] > S[i]}} W[j],$$

denote the weighted sum of numbers in S less than $S[i]$ and greater than $S[i]$ respectively. A *weighted median* of S is an entry $S[i]$ such that

$$\max(\ell_i, g_i) \leq \frac{1}{2} \sum_{j \in [n]} W[j].$$

(i) **(5 points)** Give an $O(n \log(n))$ time algorithm that, on inputs S, W, returns $i \in [n]$ so $S[i]$ is a weighted median of S.

(ii) **(15 points)** Give an $O(n)$ time algorithm that, on inputs S, W, returns $i \in [n]$ so $S[i]$ is a weighted median of S. Completing this part of the problem gives credit for the other part.

2 Problem 2

Let $n = 2^k$ for $k \in \mathbb{N} \cup \{0\}$ be a power of two. We recursively define $\mathbf{H}_k$, the $n \times n$ *Walsh-Hadamard transform* (WHT) matrix, as follows, where the base case is $\mathbf{H}_0 := 1$:

$$\mathbf{H}_k := \begin{pmatrix} \mathbf{H}_{k-1} & \mathbf{H}_{k-1} \\ \mathbf{H}_{k-1} & -\mathbf{H}_{k-1} \end{pmatrix}.$$

(i) **(5 points)** What are $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_3$?

(ii) **(5 points)** Prove that for all $k \in \mathbb{N}$, $\mathbf{H}_k^2 = C_k \mathbf{I}_n$ for some $C_k > 0$. What is C_k?

(iii) **(5 points)** Give an algorithm $\mathsf{WHT}_n(\mathbf{v})$ that, on input $\mathbf{v} \in \mathbb{R}^n$, outputs $\mathbf{H}_k \mathbf{v}$.

(iv) **(5 points)** Give an algorithm $\mathsf{WHTInv}_n(\mathbf{v})$ that, on input $\mathbf{v} \in \mathbb{R}^n$, outputs $\mathbf{H}_k^{-1} \mathbf{v}$.

3 Problem 3

Let L be an Array containing n tuples of integers (each denoting the left and right endpoints of a closed interval), and denote its i^{th} entry by $L[i] = (\ell_i, r_i) \in \mathbb{Z}^2$. Assume $\ell_i < r_i$ for all $i \in [n]$, and that no two of the $2n$ integers contained in L are equal (i.e., all interval endpoints are distinct).

For each of the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairs (i, j) with $1 \leq i < j \leq n$, exactly one of the following conditions holds for the i^{th} and j^{th} intervals, $[\ell_i, r_i]$ and $[\ell_j, r_j]$. Give an algorithm that, on input L, counts how many pairs (i, j) satisfy each in $O(n \log(n))$ time. These three numbers should sum to $\binom{n}{2}$.

- *Non-overlap.* $[\ell_i, r_i] \cap [\ell_j, r_j] = \emptyset$.
- *Containment.* Either $[\ell_i, r_i] \subset [\ell_j, r_j]$ or $[\ell_i, r_i] \supset [\ell_j, r_j]$.
- *Partial overlap.* Neither of the other two cases (non-overlap or containment) holds.

Note that you only need to solve two of these three counting problems. Correctly answering one is worth **(10 points)**, and correctly answering another is worth the other **(10 points)**.

4 Problem 4

Let L be an Array containing n real numbers, and denote its i^{th} entry by $L[i]$ for all $i \in [n]$.

(i) **(5 points)** Assume $L[i] > 0$ for all $i \in [n]$. Give an algorithm that, on input L, prints indices (i, j) with $1 \leq i \leq j \leq n$, maximizing the product $\prod_{k=i}^{j} L[k]$.

(ii) **(15 points)** Assume nothing other than $L[i] \in \mathbb{R}$ for all $i \in [n]$. Give an algorithm that, on input L, prints indices (i, j) with $1 \leq i \leq j \leq n$, maximizing the product $\prod_{k=i}^{j} L[k]$. Completing this part of the problem gives credit for the other part.

5 Problem 5

(**20 points**) Complete the assignment at this link.