Algorithms, 2023 Fall, Homework 9
(Due: Nov 30)

November 20, 2023

Problem 1 : @ Write down the pseudocode for counting the number inversion in $O(n \log n)$ time.

Problem 2 : @ KT book pp 246 problem 1

Problem 3 : @ KT book pp 246 problem 2

Problem 4 : @ Solve the recursion

$$T(n) = 3T(n/2) + n \log n.$$

You can assume $T(n) = O(1)$ for $n \leq n_0$ where n_0 is a small positive constant.

Problem 5 : KT book pp 248 problem 6.

Problem 6 : (Discrete Fourier Transformation (DFT)) In this exercise, we introduce basics about discrete Fourier transform (DFT). The sequence of N complex numbers $x_0. \dots, x_{N-1}$ is transformed into another sequence of N complex numbers $X_0, \dots, X_{N-1}$ according to the DFT formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}$$

We use $\mathbf{X} = \mathcal{F}(\mathbf{x})$ to denote this transform. The inverse discrete Fourier transform (IDFT) is given by:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{+i2\pi \frac{k}{N} n}.$$

We use $\mathbf{x} = \mathcal{F}^{-1}(\mathbf{X})$ to denote this transform.

1. Show that if we first apply DFT, then IDFT, on a sequence, we get the original sequence, i.e., $\mathcal{F}^{-1}(\mathcal{F}(\mathbf{x})) = \mathbf{x}$.

2. Prove the convolution theorem:

$$\mathcal{F}(\mathbf{x} * \mathbf{y}) = \mathcal{F}(\mathbf{x}) \cdot \mathcal{F}(\mathbf{y})$$

and

$$\mathcal{F}(\mathbf{x} \cdot \mathbf{y}) = \frac{1}{N} \mathcal{F}(\mathbf{x}) * \mathcal{F}(\mathbf{y})$$

where $*$ denotes the periodic convolution (i.e., the nth coordinate of $\mathbf{x} * \mathbf{y}$ is $\sum_{q=0}^{N-1} x_q y_{(n-q \mod N)}$) and $\cdot$ denotes the pointwise multiplication.

Bonus problems: Bonus problems are problems that may be (may be not) more challenging than other problems. You do not lose points if you do not answer bonus problems. But you will get extra points if you get correct answers for such problems. **To make your life easier, we give you two weeks for the bonus problems)**.

Problem 1 : (Necklace Alignment Problem) We are given two cyclic $\{0, 1\}$ strings X and Y, both of length n. Both contain the same number of 0s and 1s (let us say we have k 0s). Suppose the positions of 0s in X are $x_0, x_1, \ldots, x_{k-1}$ (in increasing order) and the positions of 0s in Y are $y_0, y_1, \ldots, y_{k-1}$. Without loss of generality, assume $x_0 = y_0 = 0$. We have two parameters that we want to decide. The first parameter, the offset $c \in \{0, 1, \ldots, n-1\}$ decides the clockwise rotation angle of the second string, i.e., the position of the first 0 in the second string after rotation is c. The second parameter, the shift distance $s \in \{0, 1, \ldots, k-1\}$, defines the perfect matching between the "0"s of the two strings, i.e., the ith 0 of the first string matches with the $(i+s)$th 0 of the second string. Find the offset and the shift distance of Y such that the ℓ_2 distance of X and Y (after shifting), i.e.,

$$\sum_{i=0}^{k-1} (x_i - y_{(i+s) \bmod k} - c)^2$$

is minimized. You algorithm should run in $o(n^2)$ time.