E-JUST Egypt - Japan University of Science and Technology	**Assignment 3: Spring 2021**
Department	Computer Science and Engineering
Submission due	Tue, Apr 13th, 2021 by 9:30am
Date	07/04/2021
Course Title	CSE 326: Analysis and Design of Algorithms
Instructor	Prof. Walid Gomaa
Allowed Equipment	None

Answer all of the following questions

1. (Fast multiplication of integers) Assume the problem of multiplying two n-digits integer numbers. For simplicity assume decimal notation. The standard algorithm takes $\Theta(n^2)$. A faster algorithm can be done as follows:

   - View an n digit number as two $n/2$ digit numbers concatenated, i.e. $\overline{ab}$.
   - Then multiplying the two number becomes:

   $$(10^{n/2} \cdot a + b) \times (10^{n/2} \cdot c + d) \tag{1}$$

   - Standard recursion needs computing the multiplications: ab, ac, bc, and bd which again leads to $\Theta(n^2)$.
   - However, the original problem can be expressed in terms of 3 multiplications:

   $$\begin{aligned}&(10^{n/2} \cdot a + b) \times (10^{n/2} \cdot c + d)\\ &= 10^{n/2} \times (a+b)(c+d) + 10^n ac + bd - 10^{n/2}(ac+bd)\end{aligned} \tag{2}$$

     This only needs to compute ac, bd, and $(a+b)(c+d)$.
   - So the recursive algorithm for multiplication of 2 n-digit numbers reduces to the use of 3 multiplies of $n/2$-digit numbers, and addition/subtraction.

   (a) Find the recurrence equation for this methods and solve it.

   (b) Use this method to multiply 11112222 and 33334444, both of which have 8 digits. Consider the top-level recursive call: what are the values of n, a, b, c, and d?

   (c) The top-level recursive call (multiplying 11112222 and 33334444) makes three recursive calls. This lead to three recursive calls, each computing the product of a pair of numbers that are half as long. What are these three pairs of numbers?

   (d) Suppose instead of dividing into 2 numbers of length $n/2$ we divide into 3 numbers of length $n/3$ digits (for simplicity, assume that n is a power of 3). Now suppose we can compute the overall product of a pair of n digit numbers by:

   - Multiplying 5 pairs of these $n/3$ digit numbers.
   - Add/subtract/shift the results in operations that take a total of $\mathcal{O}(n)$ time.

   State the runtime recurrence of such a divide-and-conquer algorithm, and solve it.

2. (Multiplying k n-digit numbers) We have k numbers, $A[1], A[2], ...A[k]$, each with up to n digits, with k much larger than n. Note that if we multiply one number at time, we may need to multiply an $\Omega(kn)$ digit number by an n digit number $\Omega(k)$ times, giving a total cost of at least $\Omega(k^2)$. The goal is to compute the product $A[1] \times A[2] \times \cdots \times A[k]$ more efficiently using the following two facts:

   - There is an algorithm for multiplying two numbers with n digits each in $\mathcal{O}(n^{1.6})$ time.
   - The product of k numbers each with n digits has at most $\mathcal{O}(kn)$ digits.

     (a) Show that if we have the product of $A[1] \cdots A[mid]$ and $A[mid + 1] \cdots A[k]$ where $mid = k/2$ (assume k is even), then we can compute the product of $A[1] \times \cdot \times A[k]$ in $\mathcal{O}(n^{1.6}k^{1.6})$ time.

     (b) Using the conclusion of part (a), or any other method of your choice, design an algorithm for computing the overall product of these k n-digit numbers in $O(k^{1.6}n^{1.6})$ time or better, and give a justification of its running time.

3. Develop an algorithm to find the median of two sorted arrays A and B and find the runtime complexity of your algorithm

4. Let S be a finite set of n positive integers,$S \subseteq \mathbb{Z}^+$. You may assume that all basic arithmetic operations, i.e. addition, multiplication, and comparisons, can be done in unit time.

   (a) Design an $\mathcal{O}(n \log n)$ algorithm to check the following property:

$$\forall T \subseteq S : \quad \sum_{t \in T} t \geq |T|^3 \tag{3}$$

   Argue informally that your algorithm is correct and analyze its running time.

   (b) In addition to S and n, we are given an integer k, $1 \leq k \leq n$. Design a more efficient algorithm, than that in the previous part, to check the following property:

$$\forall T \subseteq S, |T| = k : \quad \sum_{t \in T} t \geq k^3 \tag{4}$$

   Argue informally that your algorithm is correct and analyze its running time.