Problem 1. What is the exact number of atomic multiplications carried out using Karatsuba algorithm to find the product 2342×5232? Show your work.

Note: If at any step the two numbers that you're multiplying do not have the same number of digits, please use leading zeros for the number with the fewer digits to make them the same size. For example, if the multiplication is 12×3, consider it to be 12×03, but the multiplication with zero is not necessary. This is done only so that the split is nice. Also, multiplying a digit with 10 or its higher order would just be a concatenation and not count.

Problem 2. You have worked with divide and conquer algorithms and recurrences, we will try to combine it all together. In a divide and conquer algorithm, the problem is divided into smaller subproblems, each subproblem is solved *recursively*, and a *combine* algorithm is used to solve the original problem. Assume that there are a subproblems, each of size $1/b$ of the original problem, and that the algorithm used to combine the solutions of the subproblems runs in time cn^k, for some constants a, b, c, and k. For simplicity, we will assume, $n = b^m$, so that n/b is always an integer (b is an integer greater than 1). Answer the following:

(a) Write the generalized recurrence equation.

(b) Solve the recurrence equation using a recursion tree approach. Base case, $T(1) = c$.

(c) Once you obtain the solution to the recurrence equation in part(b), you will need to evaluate runtimes exactly for three cases:

  (a) $a > b^k$
  (b) $a = b^k$
  (c) $a < b^k$

  In order to help you verify your exact runtime for these three cases, the asymptotic runtimes for the three cases are given as:

$$T(n) = \begin{cases} O(n^{lg_b^a}) & \text{if } a > b^k \\ O(n^k lg_b n) & \text{if } a = b^k \\ O(n^k) & \text{if } a < b^k \end{cases}$$

Show your work.

Congratulations! you have verified a very important theorem.

Problem 3. You are given an unordered input array, $X = [x_1, x_2, \ldots, x_n]$ of integers of length, n and another sequence, $A = [a_1, a_2, \ldots, a_n]$ of distinct integers from 1 to n, such that, $a_1, a_2, \ldots, a_n$ is a permutation of $1, 2, \ldots, n$. Design an efficient algorithm and write pseudo-code / English description to order X according to the order imposed by the permutaation. In other words, for each i, $X[i]$ should appear in the position given in $A[i]$. For example, if $X = [17, 5, 1, 9]$, and $A = [3, 2, 4, 1]$ then the output should be $X = [9, 5, 17, 1]$. The algorithm should be in-place, so you cannot use an additional array.