I. Specify whether the following equations hold or not. Give an intuitive justification for your answer. Formal proofs are not needed.

1. \(3n^2 + 5n = O(n^2 \log n)\)
2. \(2^n = O(n^4 \log \log n)\)
3. \(10n^2 = O(n^2 - 100n)\)
4. \((n^2 + n)(n \log n + n^{1.1}) = O(n^3)\)
5. \(n^3 \log n = \Theta(n^3)\)
6. \(n^3 \log^5 n = O(n^{3.01})\)
7. \(4^n = \Theta(3^n)\)
8. \(n^2 \log n(\log \log n)^4 = O(n^2 \log^2 n)\)
9. \(\log^2 n = O(n^{0.05})\)
10. \(n^2 + n^{1.9} = O(n^2 + n^{0.5})\)

II. Solve the following recurrences. You can invoke a known result and write down the answer, or you can justify the answer by repeated substitutions or by induction. When needed you can assume appropriate initial values.

1. \(T(n) \leq 3T\left(\frac{n}{2}\right) + n^2\)
2. \(T(n) \leq 5T\left(\frac{n}{2}\right) + n^2\)
3. \(T(n) \leq 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{3}\right)\)
4. \(T(n) \leq 2nT(n-1)\)
5. \(T(n) \leq 2T(n-1) + T(n-2)\)

III. Let \(\delta\) be a positive-valued affinity function between pairs of symbols. Then for any pair of equal length strings \(X = a_1a_2 \cdots a_\ell\) and \(Y = b_1b_2 \cdots b_\ell\), \(\delta(X, Y) = \sum_{i=1}^{\ell} \delta(a_i, b_i)\). For any 2 strings \(X = a_1a_2 \cdots a_n, Y = b_1b_2 \cdots b_m, n \leq m, \delta(X, Y)\) is defined as

\[\max_{|Z|=n, Z \text{ a subsequence of } Y} \delta(X, Z)\]

For example, if \(\delta(a, a) = 2, \delta(a, b) = \delta(b, a) = 1, \delta(b, b) = 3\), then \(\delta(abab, bbab) = 1+3+2+3 = 9; \delta(abbaba, baabaa) = 7\), and a subsequence that maximizes the value is underlined.
Design a dynamic programming based algorithm which for a given $\delta$, $X$ and $Y$ computes $\delta(X,Y)$.

Hint: If $X = x_1 x_2 \cdots x_n$, $Y = y_1 y_2 \cdots y_m$, for any $i \leq j$, let $c[i, j] = \delta(x_1 x_2 \cdots x_i, y_1 y_2 \cdots y_j)$.

a) Specify a computation formula for $c[i, j]$,

b) Specify an order in which the entries of the $c$ matrix can be computed, and
c) What is the speed of the algorithm?

IV.

1. Assume that a set $S$ of $n$ numbers are stored in some form of balanced binary search tree; i.e. the depth of the tree is $O(\log n)$. In addition to the key value and the pointers to children, assume that every node contains the number of nodes in its subtree. Design $O(\log n)$ step algorithms for performing the following operations.

(a) Given a positive integer $k$, $1 \leq k \leq n$, compute the $k^{th}$ smallest element of $S$, and

(b) Given 2 numbers $x$ and $y$, compute the size of the subset $\{z \mid z \in S, x \leq z \leq y\}$.

2. Specify a reason why a balanced binary tree is better than a complete binary tree for storing the set $S$:

(a) A balanced binary tree is more space efficient,

(b) Inserts and Deletes can be performed faster, or

(c) Implementation of a balanced binary tree requires less RAM space