**CS 421 Algorithms (Summer 2021)**

**Homework #1 (80 points)**

**Due Date: at noon on 7/13/2021 (Tuesday).**

**This homework will be discussed during the class on 7/13/2021.**

**This homework is a preparation for 1st mid-term exam.**

**1st mid-term exam will be taken place on 7/14 (Wednesday).**

**Submission Instruction:**

- Convert your homework 1 to a single PDF file and the file name should be in a format using your name. For example, **JoeSmith421H1.pdf**
- Log into onyx and upload your homework 1 to an empty directory (i.e., the directory will contain only your homework 1 file).
- Within the directory, issue the following command
  `submit jhyeh cs421 h1`

- **Q1(10 points): Asymptotic Notations**

  (a)(3 points) Which one of the following is a wrong statement?

  1. $\Theta(n) + O(n) = \Omega(n)$
  2. $\Theta(n) + O(n) = O(n)$
  3. $\Theta(n) + \Omega(n) = \Theta(n)$
  4. $f(n) = o(g(n))$ implies $g(n) = \Omega(f(n))$

  (b)(7 points) Please use the original definition (i.e., the definition using sets) of $\Theta$-notation to show $n^2 - 1000\, n \log_{10} n = \Theta(n^2)$.

- **Q2(14 points): Divide-and-Conquer**
  Suppose that a computer does not know how to apply dynamic programming techniques to compute a function $f(n)$, but it knows how to use the `divide and Conquer` approach to compute $f(n)$ as follows. The computer takes only constant time for scalar arithmetic operations.

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ 2 \cdot f(n-1) + n & \text{if } n > 1 \end{cases}$$

  (a)(10 points) Please write down the three steps of `Divide, Conquer` and `Combine` to describe how the computer calculates $f(n)$.

  `Divide: Do nothing.`
  `Conquer:`

  `Combine:`

  (b)(4 points) Please write down the running time recurrence if $f(n)$ is computed using the above approach.

- **Q3(30 points): Recurrences**

  (a)(10 points) Given a recurrence $T(n) = 3T(n-1) + 1$, please draw the recursion tree and derive a tight bound of $T(n)$.

(b)(10 points) Given a recurrence $T(n) = 3T(n-1) + n$, please use the substitution method to verify $T(n) = O(3^n)$.
`Hint: use the hypothesis` $T(n) \leq c(3^n - n)$ `for some` $c > 0$.

(c)(10 points) Please solve the recurrence $T(n) = 3T(n-1) + n^2$ using the Master Method.
`Hint: try to transfer the equation to another form and then solve it.`

- **Q4(26 points): Dynamic programming**

  (a)(10 points) For a `Matrix-Chain` problem with 4 matrices $A_1, A_2, A_3$ and $A_4$, please construct and draw the two tables as in the book if the dimension vector for these four matrices is $< 5, 2, 5, 1, 2 >$.

  (b)(3 points) Based on the tables in (a), what is the optimal parenthesization for the product $A_1A_2A_3A_4$ ?

(c)(10 points) For a `LCS` (longest common subsequence) problem with two input sequences $X =< C, A, B, A, B, D, C >$ and $Y =< C, B, A, B, D, B, C >$, please draw the table(s) as in the book.

(d)(3 points) Based on the table(s) in (c), what is the longest common subsequence for $X$ and $Y$ ?