COMP 3270 Introduction to Algorithms

Homework 2

1. Use the Master Method to solve the following three recurrence relations and state the complexity orders of the corresponding recursive algorithms.

(a) $T(n) = 2T(99n/100) + 100n$

a = 2, b = 100/99, f(n) = 100n
$\log_{100/99}(2) > 1$, so f(n) = O(n^($\log_{100/99}(2)$ - ε)) for ε > 0
Thus T(n) = Θ(n^($\log_{100/99}(2)$))

(b) $T(n) = 16T(n/2) + n^3 lgn$

a = 16, b = 2, f(n) = n^3logn
$\log_2(16) = 4$, so f(n) = O(n^($\log_2(16)$ - ε)) for ε > 0
Thus, T(n) = Θ(n^($\log_2(16)$))

(c) $T(n) = 16T(n/4) + n^2$

a = 16, b = 4, f(n) = n^2
$\log_4(16) = 2$, so f(n) = Θ(n^($\log_4(16)$))
Thus, T(n) = Θ(n^($\log_4(16)$)*logn)

2. Use the Substitution Method to solve the following recurrence relation. Give an exact solution:

$$T(n) = T(n-1) + n/2$$

Assume that T(n) = O(n^2), i.e., T(n) = c1*n^2 + c2*n + c3 (*)
T(n-1) = c1*$(n-1)^2$ + c2*(n-1) + c3 (**)
By substituting (**) back to the recurrence relation, we have
T(n) = c1* $(n-1)^2$ + c2*(n-1) + c3 + n/2 (***)
By solving (*) and (***), we have c1 = c2 = 1/4. The solution exists, which means the assumption is correct.
Therefore, T(n) = (n^2 + n)/4 + c = O(n^2).

Order of complexity: O(n^2)

3. Use Strassen's algorithm to compute the matrix product. Show detailed procedure of your work.

$$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

$P_1 = a \cdot (f - h) = 1 \times (8 - 2) = 6$
$P_2 = (a + b) \cdot h = (1 + 3) \times 2 = 8$
$P_3 = (c + d) \cdot e = (7 + 5) \times 6 = 72$
$P_4 = d \cdot (g - e) = 5 \times (4 - 6) = -10$
$P_5 = (a + d) \cdot (e + h) = (1 + 5) \times (6 + 2) = 48$
$P_6 = (b - d) \cdot (g + h) = (3 - 5) \times (4 + 2) = -12$
$P_7 = (a - c) \cdot (e + f) = (1 - 7) \times (6 + 8) = 18$

$r = P_5 + P_4 - P_2 + P_6 = 48 - 10 - 8 - 12 = 18$
$s = P_1 + P_2 = 6 + 8 = 14$
$t = P_3 + P_4 = 72 - 10 = 62$
$u = P_5 + P_1 - P_3 - P_7 = 48 + 6 - 72 + 84 = 66$

Therefore,
$$\begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \cdot \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$$

4. Using pages 4-16 of lecture notes of Chapter 4 as a model, illustrate the operation of PARTITION on the array A = [13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11].

[13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11]
[13, 9, 19, 5, 12, 8, 7, 4, 21, 2, 6, 11]

[13, 9, 5, 19, 12, 8, 7, 4, 21, 2, 6, 11]

[13, 9, 5, 12, 19, 8, 7, 4, 21, 2, 6, 11]

[13, 9, 5, 12, 8, 19, 7, 4, 21, 2, 6, 11]

[13, 9, 5, 12, 8, 7, 19, 4, 21, 2, 6, 11]

[13, 9, 5, 12, 8, 7, 4, 19, 21, 2, 6, 11]

[13, 9, 5, 12, 8, 7, 4, 19, 21, 2, 6, 11]

[13, 9, 5, 12, 8, 7, 4, 2, 21, 19, 6, 11]

[13, 9, 5, 12, 8, 7, 4, 2, 6, 19, 21, 11]

[13, 9, 5, 12, 8, 7, 4, 2, 6, 11, 21, 19]

[11, 9, 5, 12, 8, 7, 4, 2, 6, 13, 21, 19]

i = 10

5. Show that the running time of QUICKSORT is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

When array A is sorted in decreasing order, the pivot is the smallest element. PARTITION performs a worst-case partitioning ($\Theta(n)$) and the size of one of the subproblems is only decreased by 1 (the other subproblem's size is 0). Therefore, the recursion of QUICKSORT is $T(n) = T(n-1) + T(0) + \Theta(n)$, which is $\Theta(n^2)$.