Assignment 1 :

Algorithms & Complexity

Question 1 : Algorithm Design

```
==Pseudocode For The Algorithm Finding
Every Palindromic Subset Of The String==

Initial : i = 0, j = (n - 1)
Given : string 'input'

Count(i, j)

// If the length of the input string is 2
// Just compare whether the two character is the same or not

if (j == (i + 1)) {
  if (input[i] == input[j]) { return true; }
  else { return false; }
}
```

Check Special Case

```
// Condition 1 : If i crosses j, it will be a invalid substring
// Condition 2 : If i == j that means only one character is remaining
// (require substring at least length 2)
// Both conditions need to return 0

else if (i == j || i > j) {
    return 0;
}
```

Error Situation

```
else if str[i..j] is PALINDROME {
    // Increase the count by 1
    // Check for those palindromic substring left (i, j - 1), (i + 1, j)
    // Remove common substring (i + 1, j - 1)
    return Count(i + 1, j) + Count(i, j - 1) + 1 - Count(i + 1, j - 1);
}

else {
    // If not the palindrome
    // Check for rest palindromic substrings (i, j - 1) and (i+1, j)
    // Remove common substring (i+1 , j-1)
    return  Count(i + 1, j) + Count(i, j - 1) - Count(i + 1, j - 1);
}
```

Actual Finding

Question 2 : Analyze Time Complexity

for loop : gap $\Rightarrow 2 \sim n$

for loop : starting point

$\Rightarrow 0 \sim n - gap$

Worst Case : $O(n^2)$

Best Case : $\Omega(n^2)$

Average Case : $\Theta(n^2)$

Every Case : $T(n) = n^2$

In all situations, we need to go through the whole string to check if palindromic substrings exist. Thus, time complexity is always in n^2 order of magnitude.

#

Question 3 :

Textbook Exercises 1-15 ~ 1-18 , 1-22

Ans :

1-15

$$n^2 + 3n^3 \le O(n^3)$$

for $n \ge 1$

$$n^2 + 3n^3 \le 4n^3$$

$$\Rightarrow d = 4 , N = 1$$

$$n^2 + 3n^3 \ge \Omega(n^3)$$

for $n \ge 1$

$$n^2 + 3n^3 \ge 3n^3$$

$$\Rightarrow c = 3 , N = 1$$

$$\therefore 3n^3 \le n^2 + 3n^3 \le 4n^3$$

$$c = 3 , d = 4 , n = 1$$

proof $f(n) = n^2 + 3n^3 \in \Theta(n^3)$

✱

Ans:

1-16

$$6n^2 + 20n \le O(n^3)$$

for $n \ge 5$

$$6n^2 + 20n \le 2n^3$$

$$\Rightarrow c = 2, \quad N = 5$$

$$6n^2 + 20n \ge \Omega(n^3)$$

set $c = 2$

$$6n^2 + 20n \ge 2n^3$$

The equation holds only

in $0 \le n \le 5$

From definition of Ω,

$6n^2 + 20n \notin \Omega(n^3)$

because $n \ge N$

not established

proof $6n^2 + 20n \in O(n^3)$,

$6n^2 + 20n \notin \Omega(n^3)$

#

Ans :

(1-17) By using property 7 ,

$$\boxed{\begin{array}{l} \text{If } c \geq 0 , \quad d > 0 , \\ g(n) \in O(f(n)) \text{ and} \\ h(n) \in \Theta(f(n)) , \text{ then} \\ c \times g(n) + d \times h(n) \in \Theta(f(n)) \end{array}}$$

$$5n^5 + 4n^4 + 6n^3 + 2n^2 + n + 7 \in \Theta(n^5)$$

$$\Rightarrow n^5 + 4n^5 + 4n^4 + 6n^3 + 2n^2 + n + 7 \in \Theta(n^5)$$

$$\Rightarrow 1 \times n^5 + 1 \times (4n^5 + \cdots + 7) \in \Theta(n^5)$$

$$c \times g(n) + d \times h(n) \in \Theta(f(n))$$

proof $5n^5 + 4n^4 + 6n^3 + 2n^2 + n + 7 \in \Theta(n^5)$

※

Ans :

1-18 By using property 7 ,

$$\boxed{\begin{array}{l} \text{If } c \geq 0 , \; d > 0 , \\ g(n) \in O(f(n)) \text{ and} \\ h(n) \in \Theta(f(n)) , \text{ then} \\ c \times g(n) + d \times h(n) \in \Theta(f(n)) \end{array}}$$

$$P(n) = a_k n^k + a_{k-1} n^{k-1} + \cdots + a_1 n + a_0 \in \Theta(n^k)$$

$$c \times g(n) + d \times h(n) \in \Theta(n^k)$$

$$\Rightarrow c \times n^k + \{ (a_k - c) \times n^k + a_{k-1} n^{k-1}$$

$$+ \cdots + a_1 n , a_0 \} \in \Theta(n^k)$$

for $c \geq 0$, $(a_k - c) = d > 0$

proof $a_k n^k + a_{k-1} n^{k-1} + \cdots + a_1 n + a_0 \in \Theta(n^k)$

　※

Ans:

1-22

$\Theta(\lg n) \Rightarrow (\lg n)^2, \lg(n!)$

$\Theta(n) \Rightarrow \sqrt{n}, 8n+12$

$\Theta(n \lg n) \Rightarrow n \ln n$

$\Theta(n^2) \Rightarrow 5n^2 + 7n$

$\Theta(n^k), k>2 \Rightarrow n^{\frac{5}{2}}, n^n, n^n + \ln n$

$\Theta(a^n) \Rightarrow 2^{n!}, 4^n, e^n, 10^n, 5^{\lg n}$

$\Theta(n!) \Rightarrow n!, (\lg n)!$

#