Data Structure & Algorithm

NO.

DATE / /

Chapter 2.

1) why $N \geq n_0$ needed?

If there is a positive constant M for all sufficiently large value of N, the max absolute value of $T(N)$ is at $M \times |f(N)|$.
On the other hand, if there exists a positive real number M, it is necessary to have a mininum real number n_0.

$$|T(N)| \leq M|f(N)| \quad \text{for all } N \geq n_0$$

2) Because definition of Big O is $T(N) = O(f(N))$,
means that $T(N) \leq cf(N)$ for some constant C and $N \geq n_0$.
That is, $f_1(N) = \underbrace{2}_{\text{constant}} \overset{f(N)}{N}$ and $f_2(N) = \underbrace{3}_{\text{constant}} \overset{f(N)}{N}$ have the same grow rate of N, so they both $O(N)$.

3.a) $f_1(N) = 2N$, $f_2(N) = 3N$

N double

$f_1(5) = 2 \times (5) = 10 \quad (f_1(N) = 2N)$

$f_2(5) = 3 \times (5) = 15 \quad (f_2(N) = 3N)$

constant × 2

$f_1(10) = 20 = 4 \times 5 \quad (f_1(2N) = 4N)$

$f_2(10) = 30 = 6 \times 5 \quad (f_2(2N) = 6N)$

Because they have same grow rate N, so the result of double N is only affect the constant to become twice.

3.b) $f_1(N) = 2N^2$, $f_2(N) = 3N^2$

N double

$f_1(5) = 2 \times (5)^2 = 50 \quad (f_1(N) = 2N^2)$

$f_2(5) = 3 \times (5)^2 = 75 \quad (f_2(N) = 3N^2)$

constant × 4

$f_1(10) = 2 \times (2 \times 5)^2 = 8 \times (5)^2 = 200 \quad (f_1(2N) = 8N^2)$

$f_2(10) = 3 \times (2 \times 5)^2 = 12 \times (5)^2 = 300 \quad (f_2(2N) = 12N^2)$

Because they have same grow rate N^2, so the result of double N is only affect the constant to become 4 times larger than before.

NO.

DATE / /

4) Because the most important thing in Algorithm analyze is running time. We have to measure Best case, and worst-case, or average-case. Typically we analyze worst-case performance, and Big-O tells us the limit of poor performance.

5) 2^n (exponential) v.s $n!$ (factorial)

$n!$ eventually grows faster than 2^n, because $n!$ with an increased base, and 2^n with an constant base.

ex:

n	1	2	3	4	5
2^n	2	4	8	16	32
$n!$	1	2	6	24	70

$\rightarrow O(n!) > O(2^n)$ #

6.a) $4n^5 + 3n^2 - 2 \rightarrow O(n^5)$

6.b) $5^n - n^2 + 19 \rightarrow O(5^n)$

6.c) $(3/5) \times n \rightarrow O(n)$

6.d) $3n \times \log n + 11 \rightarrow O(n\log n)$

6.e) $[n(n+1)/2 + n]/2 = \dfrac{n^2 + n + 2n}{4} \rightarrow O(n^2)$

7.

```
for (int i=0 ; i < N ; i++)   //(1+N)+(N-1+1)
    System.out.println(i+1);   // 2N
```

$\rightarrow 1 + N + N + 2N = 4N + 1 \rightarrow O(N)$ #

SEASON

NO.

DATE / /

8)
```
for ( int i=0 ; i< N ; i++ )
    for ( int j=0 ; j < N ; j++ )
        system.out.println( (i+1) * (j+1) );
```

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 1 = \sum_{i=0}^{N-1} (N-1-0+1) = \sum_{i=0}^{N-1} N$$

$$\left(\text{use } \sum_{i=1}^{N} i = \frac{n(n+1)}{2} \right) = 1 + \frac{(N-1)(N-1+1)}{2} = \frac{N(N-1)}{2} \rightarrow O(N^2)$$ #

9)
```
for ( int i=0 ; i < N+1 ; i++ )          // 1 + (N+2) + (N+1)
    for ( int j=0 ; j < 2N ; j++ )       // 1 + (2N+1) + (2N-1+1)
        System.out.println( (i+1)*(j+1) );   // 4 unit + (2N)
```

$$\sum_{i=0}^{N} \sum_{j=0}^{2N-1} 1 = \sum_{i=0}^{N} (2N-1-0)+1 = \sum_{i=0}^{N} 2N$$

$$= \frac{2N(2N+1)}{2} + 1 = \frac{4N^2+2N+2}{2} = 2N^2+N+1$$

$$\rightarrow O(N^2)$$ #

10)
```
if ( num < N ) {
        for ( int i=0 ; i < N ; i++ )    // 1 + (N+1) + (N-1+1)
            system.out.println(i);        // N
}
else
    System.out.println("too many");  // 1 unit
```

$\Rightarrow (1+N+N) + N + 1 = 3N+2$

$\Rightarrow O(N)$ #

SEASON

NO.

DATE / /

$* \log^k N = O(N)$

11)
```
int i=N;
while (i>0){  // logN
    i = i/2      // 2 unit * logN
}
```

ex: $i=4$
$i = 4/2 = 2$
$i = 2/2 = 1$
$i = 1/2 = 0$

$\rightarrow \log N + 2\log N$ take 2 step, $\log_2 4 = 2$.

$\rightarrow 3\log N \rightarrow O(N)$ #

? 12)
```
public static int div (int numItems){
    if(numItems == 0)
        return 0;           // 1
    else
        return numItem % 2 + div(numItem/2);
}
```

$\Rightarrow T(N) = a + T(N/2)$

$\Rightarrow T(N) = a * \log_2 N + T(0) \Rightarrow T(N) = a * \log_2(N) + b$

$\Rightarrow O(\log N)$ #

ex: numItems = 8

$\overset{T(N)}{T(8)} \rightarrow a + \overset{T(N/2)}{T(4)} = a + (2a+b) = 3a+b$

$T(4) \rightarrow a + T(2) = a + (a+b) = 2a+b$

$T(2) \rightarrow a + T(1) = a + b = a+b$

Assume $T(1)$ & $T(0)$ = constant

$T(1) \rightarrow a + T(0) = b$

$T(0) = 0$

$\Rightarrow T(N) = a \times (\log N) + b$