1. Finding Complexities:

a) $O(n^2) \cdot O(n) = \underline{O(n^3)}$

As shown in the nested-loops example in the notes, if j is dependent on i, then the complexity is:

$\sum_{i=1}^{n} \frac{n(n+1)}{2} = O(n^2)$. In this case one of the 'n' values is in fact 'n^2' $\therefore \frac{n^2(n+1)}{2} = \frac{n(n^2+1)}{2} = O(n^3)$

b) $O(n^2 + \log_2 n) = O(n^2)$.

This exercise is exact to the one in the lecture notes.

1 nested loop : $\sum_{i=1}^{n} \frac{n(n+1)}{2} = O(n^2)$
1 while loop : $O(\log_a n)$, where a is the divider of k.

$\therefore O(n^2) + O(\log_2 n) = O(n^2 + \log_2 n) = O(n^2)$

c) Since there are 2 loops nested in the first while-loop, the complexity is : $O(n(n+1)) = O(n^2+n)$ since

```
k = m
while k > m:
    # Statements
    k -= 1
```

is done once. ie. $O(1)$.

Next while loop has complexity $O(n)$

$\therefore O(n^2+n) + O(n) = O(n^2+2n) = O(n^2)$

d) Last else statement: Complexity: $\underline{O(5)}$ as something is done 5 times.

Elif: Since $c < 5$, the space between them can go from $1 \rightarrow n$. $\therefore$ $\max(1, n) = n$, so the complexity is $\underline{O(n)}$

If: for loop in range(a) has complexity $\underline{O(n)}$ as 'a' can any size.
add to that complexity $\underline{O(n)}$ for the "for k in range(b)" loop, for the same reasons.

$\therefore$ Total complexity: $O(n) + O(n) + O(n) + O(5)$
$\Rightarrow O(3n) + O(5)$
$\Rightarrow O(3n+5) = O(3n) = \underline{O(n)}$.

2. Unfolding and Proofing:

a) $T(n) = 100(T(\frac{n}{10})) + n^2$

$$= 100\left(100T\left(\frac{n}{100}\right) + \frac{n^2}{100}\right) + n^2$$
$$= 100^2 T\left(\frac{n}{100}\right) + n^2 + n^2$$
$$= 100^2 T\left(\frac{n}{100}\right) + 2n^2$$

$$= 100^2\left(100T\left(\frac{n}{1000}\right) + \frac{n^2}{100^2}\right) + 2n^2$$
$$= 100^3 T\left(\frac{n}{1000}\right) + n^2 + 2n^2$$
$$= 100^3 T\left(\frac{n}{1000}\right) + 3n^2$$

$$= 100^3\left(100T\left(\frac{n}{10{,}000}\right) + \frac{n^2}{1000^2}\right) + 3n^2$$
$$= 100^4 T\left(\frac{n}{10^4}\right) + n^2 + 3n^2$$
$$= 100^4 T\left(\frac{n}{10^4}\right) + 4n^2$$

...

...

...

$\therefore T(n) = 100^i T\left(\frac{n}{10^i}\right) + in^2$

When $i = \log_{10}(n)$, $T\left(\frac{n}{10^i}\right) = T(1) = 1$

$\therefore 100^{\log_{10}(n)}(1) + \log_{10}(n)(n^2)$

$\therefore 10n + n^2\log_{10}(n) = T(n)$

Proof: $10n + n^2\log_{10}(n) \leq 11n^2\log_{10}(n)$

$10n \leq 10n^2\log_{10}(n)$

$1 \leq n\log_{10}(n)$

$\therefore T(n) = O(10n + n^2\log_{10}(n)) = \underline{O(n\log_{10}(n) + n)}$

b) Guess : $T(n) = O(10n + \log_{10}(n)(n^2)) \Rightarrow \Theta(n \log_{10}(n) + n)$

Proof: Assumption : $T(n) = n \log_{10}(n) + n$

Base : $T(1) = 1^2 \log_{10}(1) + 1$
$= 0 + 1$
$T(1) = 1$

Step: $T(n) = 100T(\frac{n}{10}) + n^2$
$= 100(\frac{n}{10} \log_{10}(\frac{n}{10}) + \frac{n}{10}) + n^2$
$= 10n(\log_{10}(n) - \log_{10}(10)) + 10n + n^2$
$= 10n \log_{10}(n) - 10n + 10n + n^2$

$= 10n \log_{10}(n) + n^2$

3. Master Theorem:

a) $T(n) = 8T(\frac{n}{2}) + n^3$

$a = 8$
$b = 2$
$f(n) = n^3$
$n^{\log_b a} = n^{\log_8(2)} = \sqrt[3]{n} = n^{\frac{1}{3}}$

$f(n) = \Omega(n^{\log_a b + \varepsilon})$
$n^3 = \Omega(n^{\frac{1}{3} + \frac{8}{3}})$
and
$\frac{n^3}{8} \leq \frac{1}{8} n^3$, where $\varepsilon = \frac{8}{3}$ and $c = \frac{1}{8}$

$\therefore T(n) = \Theta(n^3)$

b) $T(n) = T(\frac{n}{2}) + n\log n$

$a = 1$
$b = 2$
$f(n) = n\log n = \log(n^n)$
$n^{\log_b a} = n^0 = 1$

$f(n)$ grows faster than 1, therefore $f = \Omega(g)$

$n\log n = \Omega(n^{(\log_b a) + \varepsilon})$ where $\varepsilon > 0$.
Let $\varepsilon = \log_n(n\log n)$: $n\log n = \Omega(n^{0+\varepsilon}) = \Omega(n\log n)$
and
$\frac{n}{2}\log(\frac{n}{2}) \leq c(n\log n) \Rightarrow \frac{n\log n - n\log 2}{2} \leq \frac{1}{2}(n\log n)$, where $c = \frac{1}{2}$

$\therefore T(n) = \Theta(n\log n)$

c) $T(n) = 3T(\frac{n}{3}) + \log n$

$a = 3$
$b = 3$
$f(n) = \log n$
$n^{\log_b a} = n^1 = n$

$f(n)$ grows slower than n, therefore $f = O(g)$

$\log n = O(n^{(\log_b a) - \varepsilon})$, where $\varepsilon > 0$

Let $\varepsilon = 1 - \log_n(\log n)$

Then: $\log n = O(n^{1-(1-\log_n(\log n))})$
$= O(n^{\log_n(\log n)})$
$\log n = O(\log n)$

$\therefore T(n) = \Theta(n)$