S-2

a) $f(n) = \frac{n^2}{150} = \Omega(n)$

$f(n) \geq c \cdot g(n)$

$\frac{n^2}{150} \geq c \cdot n$

$= \frac{n}{150} \geq c$

$n = 150 \rightarrow \frac{150}{150} \geq c \Rightarrow 1 \geq c$

$c = 1$

b) $f(n) = 5n^2 + 8n + 15 = O(n^2)$

$f(n) \leq g(n)$

$5n^2 + 8n + 15 \leq cn^2$

$5n^2 - cn^2 + 8n + 15 \leq 0$

$n = 1 \mid 5 - c + 8 + 15 \leq 0$ ie

$c \geq 28$

c) $f(n) = 7n^2 - 3n + 2 = O(n)$

$f(n) \leq g(n)$

$7n^2 - 3n + 2 \leq cn$

$7n^2 - 3n - cn + 2 \leq 0$

$n = 1 \rightarrow 7 - 3 - c + 2 \leq 0$

$c \geq 6$

d) $f(n) = 5n^3 + 2n^2 + 4n + 6 = \Omega(n^3)$

$c_1 \cdot n^3 \leq 5n^3 + 2n^2 + 4n + 6 \leq c_2 n^3$

$c_1 = 5$

$c_2 = 6$

$5n^3 \leq 5n^3 + 2n^2 + 4n + 6 \leq 6n^3$

$n \geq 4$

8-3 $T(n) = \begin{cases} 1 & n=1 \\ T(\frac{n}{2}) + T(\frac{n}{2}) + 1 & n>1 \end{cases}$

$T(n) = 2T(\frac{n}{2}) + 1$

$n = \frac{n}{2}$; $T(\frac{n}{2}) = 2T(\frac{n}{4}) + 1$

$T(n) = 2\left[2T(\frac{n}{2^2}) + 1\right] + 1$

$= 2^2 T(\frac{n}{2^2}) + 2 + 1 \quad n = \frac{n}{2} \Rightarrow T(\frac{n}{2}) = 2T(\frac{n}{2^3}) + 1$

$= 2^2 T(2T(\frac{n}{2^3}) + 1] + 2 + 1$

$= 2^3 T(\frac{n}{2^3}) + 2^2 + 2 + 1 \Rightarrow n = \frac{n}{2^3} \Rightarrow 2T(\frac{n}{2^4}) + 1$

$= 2^3 T(2T(\frac{n}{2^4}) + 1) + 2^2 + 2 + 1$

$= 2^4 T(\frac{n}{2^4}) + 2^3 + 2^2 + 2 + 1$

$= 2^5 T(\frac{n}{2^5}) + 2^4 + 2^3 + 2^2 + 2 + 1$

$= 2^6 T(\frac{n}{2^6}) + 2^5 + 2^4 + 2^3 + 2^2 + 2 + 1$

$\Rightarrow 2^k T(\frac{n}{2^k}) + 2^{k-1} + 2^{k-2} + \dots + 2 + 2^0$

$\frac{n}{2^k} = 1 \quad 2^k = n \Rightarrow k = 6g_2^n$

$2^0 + 2^1 + 2^3 + \dots 2^{k-1} = 2^{k+1} - 2^k - 1 \Rightarrow 2^k - 1$

$T(1) = 1 \quad n\,T(1) + 2^k - 1$

$n \cdot 1 + n - 1 \Rightarrow T(n) = 2n - 1 = O(n)$

S-4

①
```
for (i=0; i<n; i++)
    for (j=0; j<n*n; j++)
        print ("*");
```

②
```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        print ("*");
```

③
```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        for (k=0; k<j; k++)
            print ("*");
```

④
```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        for (k=0; k<j; k++)
            print ("*");
```

Code Fragment	Big-O	Number of * asterisks
1	$O(n^3)$	n^3
2	$O(n^2)$	$\frac{n^2-n}{2}$
3	$O(n^3)$	$\frac{n(n-1)\cdot(n-2)}{6}$
4	$O(n^3)$	$\frac{n^2(n-1)}{2}$

3-5

25 — 80 — 52 — 36 — 1

Bubble Sort

iteration 1

Step 1 → 25 80 52 36 1 do not swap

Step 2 → 25 80 52 36 1 swap

Step 3 → 25 52 80 36 1 swap

Step 4 → 25 52 36 80 1 swap

Step 5 → 25 52 36 1 80

iteration 2

Step 1 → 25 52 36 1 80 do not swap

Step 2 → 25 52 36 1 80 swap

Step 3 → 25 36 52 1 80 swap

Step 4 → 25 36 1 52 80 do not swap

Step 5 → 25 36 1 52 80

iteration 3

Step 1 → 25 36 1 52 80 do not swap

Step 2 → 25 36 1 52 80 swap

Step 3 → 25 1 36 52 80 do not swap

Step 4 → 25 1 36 52 80 do not swap

Step 5 → 25 1 36 52 80

iteration 4

Step 1 → 25 1 36 52 80 swap

Step 2 → 1 25 36 52 80 do not swap

=> 1 25 36 52 80 => final Sort

Insertion sort

25 — 80 — 52 — 36 — 1

iteration 0 → 25 80 52 36 1
iteration 1 → 25 80 52 36 1
iteration 2 → 25 52 80 36 1
iteration 3 → 25 36 52 80 1
iteration 4 → 1 25 36 52 80 ← final sorted Array

Selection Sort

iteration 0 (25) - 80 - 52 - 36 - (1)

iteration 1 1 80 52 36 25 (swap(25,1
iteration 2 1 25 52 36 80 (swap(80,2
iteration 3 1 25 36 52 80 (swap(52,36

final sorted array