Question 1: State the runtime requirements in big-O for each of t he following code fragments. [/1.0]

	Code Fragment	Running Time in big-O

a	`void f(int n) {`
	`  for(int i=0; i < n; i++) {`
	`    for(int j=0; j < 10; j++) {`
	`      for(int k=0; k < n; k++) {`
	`        for(int m=0; m < 10; m++) {`
	`            System.out.println("!");`
	`  } } } }`
	`}`
		O(n2)

b	`int a = 0, i = N;`
	`while (i > 0) {`
	`    a += i;`
	`    i /= 2;`
	`}`
		O(log n)

Question 2: Give the best Big-O characterization for each of the following running time estimates F(*n*), where *n* is the size of the input problem. [/3.0]

F(n)	Big-O Characterization of F(n)
$1+2+\ldots+(n-2)+(n-1)+n$	**$O(n^2)$**
$\log(7n^2)$	**O(log n)**
$\log 2^n$	**O(n)**
$n^3(1+6n+78n^2)$	**$O(n^5)$**
$2^{10}+1000$	**O(1)**
$100\ n \log n + 2 \log n$	**O(nlogn)**

Question 3: Use the definition of big-O to prove that: [/2.0]

$n^3 + 20n$ is $O(n^3)$

from the definition of big-O

n^3+20n is $O(n^3)$ if $n^3+20n <= c.\ n^3$ for all $n<=n0$

assume c = 2,

$n^3+20n <= 2.\ n^3$

$20n <= n^3$

divide over n

$20<=n^2$ true for $n>=5$

So, n^3+20n is $O(n^3)$ for c=2 and n>=5

Another possible solution:

from the definition of big-O

n^3+20n is $O(n^3)$ if $n^3+20n <= c.\ n^3$ for all $n<=n0$

$(1+20)\ n^3 <= c.\ n^3$

Then, n^3+20n is $O(n^3)$ for c = 21 and n>= 1

	Statements	S/E	Freq.	Total
1	`public void func2(int n) {`	0	-	-
2	`for (int i = 0; i < n * n; i++) {`	1	$n^2 + 1$	$n^2 + 1$
3	`System.out.println(i);`	1	n^2	n^2
4	`for (int j = 2 * n; j > n; j--)`	1	$n^2(2n - n + 1)$	$n^3 + n^2$
5	`System.out.println(j);`	1	$n^2(2n - n)$	n^3
6	`}`	0	-	-
7	`System.out.println("Goodbye!");`	1	1	1
8	`}`	0	-	-
	Total	$2n^3 + 3n^2 + 2$		
	Big Oh	$O(n^3)$		

Problem 3

Analyze the performance of the following algorithms theoretically:

	Statements	S/E	Freq.	Total
1	`public void func1(int n) {`	0	-	-
2	`for (int i = 0; i < n * log(n); i++) {`	1	$n \log n + 1$	$n \log n + 1$
3	`System.out.println(i);`	1	$n \log n$	$n \log n$
4	`for (int j = 2; j < n; j++)`	1	$n \log n\,(n - 2 + 1)$	$n^2 \log n - n \log$
5	`System.out.println(j);`	1	$n \log n\,(n - 2)$	$n^2 \log n - 2n \log$
6	`}`	0	-	-
7	`System.out.println("Goodbye!");`	1	1	1
8	`}`	0	-	-
	Total	$2n^2 \log n - n \log n + 2$		
	Big Oh	$O(n^2 \log n)$		

```
(1)  sum = 0;
     for( i = 0; i < n; i++ )
         sum++;

(2)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < n; j++ )
             sum++;

(3)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < n * n; j++ )
             sum++;

(4)  sum = 0;
     for( i = 0; i < n; i++ )
         for( j = 0; j < i; j++ )
             sum++;
```

[True/False] The function f(n) = n log n is O(log n).

- a. False
- b. True

Order the following functions by asymptotic growth rate.

4n log n +2n	**2^{10}**
3n + 100 log n	**4n**
n^2 + 10n	**2^n**
n^3	**n log n**

Question 3: Express the running time of the following functions in big-O notation. [0.5 /] 0.25 point each.

1) 2^{10}
2) $\log n$
3) n
4) $n \log n$
5) n^2
6) n^3
7) 2^n

F(n)	Big-O Characterization of F(n)
1000 nlog n + logn	$O(n \log n)$ ✓
2^{10}	$O(1)$ ✓

Question 4: Use the definition of big-O to prove that: $9n^3 + 2n$ is $O(n^3)$ [1.0/ 1]

$f(n) \leq C \cdot n^3 \quad n \geq n_0$ $\qquad f(n) \leq C \cdot (g(n)) \quad n \geq n_0$

assume $C = 10$

$9n^3 + 2n \leq 10n^3 - 9n^3$

$2n \leq n^3$

~~f(n) is big~~ $f(n)$ is $O(n^3)$ for $c = 10$, $n_0 = 2$ ✓

Question 2: Give the best Big-O characterization for each of the following running time estimates F(n), where n is the size of the input problem. [2.0/2]

1) 2^{10}
2) $\log n$
3) n
4) $n \log n$
5) n^2
6) n^3
7) 2^n

F(n)	Big-O Characterization of F(n)	Score
$\log(n) + 10000$	$O(\log n)$ ✓	[0.5/0.5]
$2^{10} + 3^{10}$	$O(1)$ ✓	[0.5/0.5]
$1+2+\ldots+(n-2)+(n-1)+n$	$O(n^2)$ ✓	[0.5/0.5]
$n \log n + 15n + 0.002\, n^2$	$O(n^2)$ ✓	[0.5/0.5]

Question 3: Use the definition of big-O to <u>prove</u> that: [2.0/2.5], 1 point each.

1. 2^{n+2} is $O(2^n)$

   Hint: $2^{a+b} = 2^a \times 2^b$

$2^{n+2} \leq c \cdot 2^n$ for $n \geq n_0$

0.5

assume $c=3$

$2^{n+2} \leq 2^{n+3}$?

0

$2^n \cdot 2^2 \leq 2^n \cdot 2^3$ ⇝ it's $O(2^n)$ for $c=3$, $n_0=1$

$2^n \cdot 2^2 \leq 2^n \cdot 2^3$

$2 \cdot 4 \leq 2 \cdot 8$

$8 \leq 16$

2. $4n^3 + n^2$ is $O(n^3)$

$4n^3+n^2 \leq cn^3$ for $n \geq n_0$

1

assume $c=6$

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

$n^2 \leq 2n^3$

it's $O(n^3)$ ~~True~~ for $c=6$, $n_0=1$

Q4. [3 points] Consider the following code fragments/algorithm in the table below. For each, state the runtime of the algorithm in **big-Oh notation**.

No.	Algorithm	Runtime expressed in big-Oh

1	`//N is a large number`
	`int sum = 0;`
	`for (int n = N; n > 0; n -= 2)`
	`    for(int i = 0; i < n; i++)`
	`        sum++;`
		$O(n^2)$

2	`//N is a large number`
	`int sum = 0;`
	`for (int i = 1; i < N; i ++)`
	`    for (int j = 0; j < 10; j++)`
	`        sum++;`
		$O(n)$

3	**Algorithm** Algo (k) **Input:** k , a positive integer **Output:** k-th even natural number (the first even being 0) **if** (k = 1) **then return** 0 **else** **return** Algo (k-1) + 2	**O(k) or is k<=n then O(n)**

Q1 [2 points] Give the Big-Oh notation for the following functions

$f(N) = N^2 + \log N^2 + 2N \log N$	$O(N^2)$
$f(N) = (N \cdot (100N + 5000000))^2$	$O(N^4)$
$f(N) = N^{1/2} + \log(\log N)$	$O(N)$
$f(N) = 1000^{100} + \log N$	$O(\log N)$

Q3 [4]. Given the following code, estimate the number of operations and describe the worst case running time in Big-Oh notation in terms of the variable n. Show trace where appropriate.

```
public int Mul(int [] A, int i, int t)
{
   if (i >= A.length-1)
     return t;
   else
     return Mul(A, ++i, t * A[i]);
}
```

Mul() is recursively called i times. I would run up to n, which is the length of this array. The run time will be O(n)

```
Mul(new int[] {1,2,3,4}, 0, 1)
```

24 Mul(A, 0, 1)

24 Mul(A, 1, 1*2)

24 Mul(A, 2, 2*3)

24 Mul(A, 3, 6*4)

24 Mul(A, 4, 24)

```
public void diag(int n) {
  int count = 0;
  for (int i = 0; i < n; i++)
    for (int j = 0; j < i; j++)
      if(i==j)
        count++;
}
```

The if statement will execute n/2 * (n+1) times

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

which is $O(n^2)$

```
diag(5)

i:0 (1 times)
j=0

i:1 (2 times)
j=0
j=1

i:2 (3 times)
j=0
j=1
j=2

i:3 (4 times)
j=0
j=1
j=2
j=3

i:4 (5 times)
j=0
j=1
j=2
j=3
j=4
```

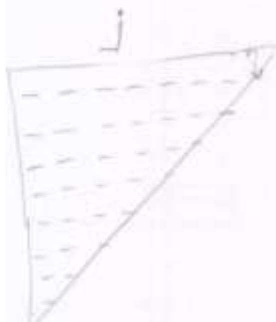
Question 2. (2 + 3 + 1 + 1 + 6 = 13 marks)

(a) How many times is the **count++** executed in the following code segment? [2 marks]

```
count = 0;
for(int i = 1; i < 10; i++) {
    count++;
}
```

9-times

(b) What is the runtime for the following code snippet? Give the runtime as an equation $T(n)$ based on estimation and the Big-Oh notation. [3 marks]




```
1. s = 0;
2. for (int i = 0; i < n; i++)
3.   for (int j = i; j < n; j++)
4.       s = s + 1;
```

1 op
$1 + 2n$ ops
$(1+2n)*n$ ops
2 ops

is in a nested loop.

Worst Case Scenario; line 3 executes n times;
in reality the nested loops in line 2 and 3 run $\frac{1}{2}n^2$ times.

Estimation

$$T(n) = 2n^2 + 3n + 4$$

which is $O(n^2)$

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 i = 1;
2 while(i < n)
3     i = i * 2;
```

1. Line 1: (A) 0 (B) 1 (C) 2 (D) n (E) n^2
2. Line 2: (A) n (B) $n+1$ (C) $\log(n)$ (D) $\log(n)+1$ (E) 2^n
3. Line 3: (A) n (B) $n+1$ (C) $\log(n)$ (D) $\log(n)+1$ (E) 2^n
4. Total O: (A) 1 (B) n (C) n^2 (D) $\log(n)$ (E) 2^n

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 c = 10;
2 for (i = 1; i <= c; i++)
3   for (j = 0; j < n; j++)
4     count++;
```

1. Line 1: (A) 0 (B) 1 (C) 2 (D) n (E) n^2
2. Line 2: (A) n (B) c (C) 11 (D) 10 (E) 9
3. Line 3: (A) n (B) $10n$ (C) $10(n+1)$ (D) c (E) n^2
4. Line 4: (A) $count+2$ (B) $10n$ (C) $11n$ (D) n^2 (E) $n(n+1)/2$
5. Total O: (A) 1 (B) n (C) n^2 (D) $n\log(n)$ (E) n^3

(c) Choose the correct answer:

1. $n^3+n^2\log n$ is : (A) $O(n^3)$ (B) $O(n^2)$ (C) $O(n^2\log(n))$ (D) $O(n^5)$ (E) None
2. 2^n+n^n is : (A) $O(n)$ (B) $O(n^2)$ (C) $O(2^n)$ (D) $O(n^n)$ (E) None
3. $n^4\log n+2^n$ is : (A) $O(n)$ (B) $O(n^4)$ (C) $O(n^5)$ (D) $O(\log(n))$ (E) **None**

4. When traversing all nodes in a binary tree of depth d. The complexity would be:

(A) $O(d)$ (B) $O(d^2)$ (C) $O(2^d)$ (D) $O(\log(d))$ (E) None

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 sum = 1;
2 for (i = 1; i <= n; i++) {
3   sum+= i;
4   for (j = i; j >= 2; j--)
5     sum--;}
```

1. Line 1: (A) 1 (B) 2 (C) 3 (D) n (E) $2n$
2. Line 2: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
3. Line 3: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
4. Line 4: (A) n^2 (B) $n(n-1)/2$ (C) $(2n+1)/2$ (D) $(2n-1)/2$ (E) $n(n+1)/2$
5. Line 5: (A) n^2 (B) $n(n-1)/2$ (C) $(2n+1)/2$ (D) $(2n-1)/2$ (E) $n(n+1)/2$
6. Total O: (A) 1 (B) n (C) n^2 (D) $n\log(n)$ (E) n^3

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 count = 0;
2 for (i = 1; i < n+1; i++)
3   count ++;
4 for (j = 0; j <= count; j++)
5   k = j+1;
```

1. Line 1: (A) 0 (B) 1 (C) 2 (D) n (E) n^2
2. Line 2: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
3. Line 3: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
4. Line 4: (A) $count+2$ (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n(n+1)/2$
5. Line 5: (A) $count+1$ (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n(n-1)/2$
6. Total O: (A) 1 (B) n (C) n^2 (D) $n\log(n)$ (E) n^3

(c) Choose the correct answer:

1. $n^2 + n\log n^4$ is : (A) $O(n)$ (B) $O(n^2)$ (C) $O(n\log(n))$ (D) $O(n^4)$ (E) None
2. $n^2 + 1000n$ is : (A) $O(n)$ (B) $O(n^2)$ (C) $O(n\log(n))$ (D) $O(nn^2)$ (E) None
3. $n^4\log n + n!$ is : (A) $O(n!)$ (B) $O(n^4)$ (C) $O(n^5)$ (D) $O(\log(n))$ (E) None
4. Algorithm A is $O(n)$, and Algorithm B is $O(2n)$. Given the same input:

   (A) A always finishes before B. (B) B always finishes before A. (C) A and B finish at the same time. (D) B requires double the time taken by A. (E) None

(c) Which of the following two algorithms has a better time complexity:

(i) algorithm A with a step count function $2^{100} + \log n^{100}$

(ii) algorithm B with step count function $n + 2\log n$. [1 mark]

→ Corred.

(d) Which of the following two algorithms has a better time complexity:

(i) algorithm A with a growth rate $O(n^2)$

(ii) algorithm B with a growth rate $O(nlogn)$. [1 mark]

Corred.

(e) For what values of c and n_0, the function is $O(n^3)$, $g(n)$ is $O(n^2 \log(n))$.([6 marks]

(i) $f(n) = 4n^3 + 6n^2 + 2n + 1$

$4n^3 + 6n^2 + 2n + 1 < cn^3$

$1 < n^3(c-4) - 6n^2 - 2n$

Assume $c = 5$ and $n_0 = 10$ then

Yes

$1 < 1000 - 600 - 20$ which is true hence $f(n)$ is $O(n^3)$

(ii) $g(n) = n^2 \log(10n^4 + 7) - 3n$

$n^2 \log(10n^4+7) - 3n + 0 \leq cn^2 \log n$

$0 \leq cn^2 \log n - n^2 (\log(10n^4+7)) + 3n$

NO

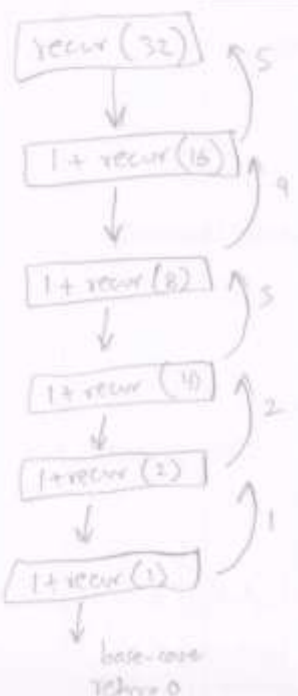
$0 \leq n^2 (c\log n - \log(10n^4+7)) + 3n$

which ever value of n, this will always give -ve time ; hence the statement is wrong.

(b) Show the trace for the following recursive method; estimate the run-time complexity as given an appropriate Big-Oh notation. For tracing use the top-level call: **recur(32)** [3.5 marks]

base case
return 8.

```
int recur (int n) {
   if (n == 1)                    ——— 1 op
      return 0;                   ——— 1 op
   return 1 + recur(n / 2);       ——— 1+2 ops
}
```



returns 5.

estimated runtime
because of division by2 we get log

$T(n) = 4 \log n + c$

which is $O(\log n)$

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int A = 0;
2 for (int i = 1; i <= n; i++)
3   for (int j = 0; j < i; j++)
4     A++;
```

1. Line 1: (A) 0 (B) 1 (C) 2 (D) n (E) A
2. Line 2: (A) A (B) i (C) $i+1$ (D) n (E) $n+1$
3. Line 3: (A) n^2 (B) $n(n+1)/2$ (C) $n(n+1)/2+1$ (D) $(n^2+3n)/2$ (E) $n(n-1)/2-1$
4. Line 4: (A) A^2 (B) n^2 (C) $(n^2+3n)/2$ (D) $n^2(n+1)/2+1$ (E) $n(n+1)/2$
5. Tightest Total O: (A) n (B) n^2 (C) n^3 (D) n^4 (E) None

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int i = 1;
2 while (i < n) {
3   i++;
4   if (i > 7) break;
5 }
```

1. Line 1: (A) 1 (B) 0 (C) i (D) n (E) $n+1$
2. Line 2: (A) 8 (B) 7 (C) n (D) $n-1$ (E) $n+1$
3. Lines 3 (and similarly 4): (A) n (B) $n-1$ (C) 6 (D) 7 (E) 8
4. Tightest Total O: (A) 1 (B) n (C) $\log(n)$ (D) n^2 (E) 2^n

(c) Choose the correct answer:

1. $n^7+n^4+n^2+\log n$ is : (A) $O(n^2)$ (B) $O(n^4)$ (C) $O(n^7)$ (D) $O(\log(n))$ (E) None
2. $2^n+n!$ is : (A) $O(n^2)$ (B) $O(2^n)$ (C) $O(n!)$ (D) $O(n^n)$ (E) None
3. $n+\log n^3+6$ is : (A) $O(n)$ (B) $O(\log n^3)$ (C) $O(n\log n)$ (D) $O(n^3)$ (E) None
4. The time complexity of inserting an element in a heap of n elements is:

   (A) $O(n^2)$ (B) $O(n)$ (C) $O(2^n)$ (D) $O(\log(n))$ (E) None

Q1 (a) [2 points]. Sam gives the run-time for an algorithm using function $f(x)$. Prove, for what values of n_0 and constant c, $f(x)$is $O(n^3)$.

$$f(n) = 2x^3 + 5x^2 + 12$$

$$2n^3 + 5n^2 + 12 \leq cn^3$$

$$12 \leq cn^3 - 2n^3 - 5n^2$$

$$12 \leq (c-2)\, n^3 - 5n^2$$

set $n_0 = 1$

$c = 20$

$$12 \leq (20-2) \times 1 - 5 \times 1$$

$$12 \leq 18 - 5$$

$$12 \leq 13$$

so $f(x)$ is $O(n^3)$ for $c = 20$, $n_0 = 1$

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 sum = 1;
2 for (i = 1; i <= n; i++) {
3   sum+= i;
4   for (j = i; j >= 2; j--)
5     sum--;}
```

1. Line 1: (A) 1 (B) 2 (C) 0 (D) n (E) $2n$
2. Line 2: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
3. Line 3: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
4. Line 4: (A) n^2 (B) $n(n-1)/2$ (C) $(2n+1)/2$ (D) $(2n-1)/2$ (E) $n(n+1)/2$
5. Line 5: (A) n^2 (B) $n(n-1)/2$ (C) $(2n+1)/2$ (D) $(2n-1)/2$ (E) $n(n+1)/2$
6. Total O: (A) 1 (B) n (C) n^2 (D) $n\log(n)$ (E) n^3

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 count = 0;
2 for (i = 1; i < n+1; i++)
3   count ++;
4 for (j = 0; j <= count; j++)
5   k = j+1;
```

1. Line 1: (A) 0 (B) 1 (C) 2 (D) n (E) n^2
2. Line 2: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
3. Line 3: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n-2$
4. Line 4: (A) $count+1$ (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n(n+1)/2$
5. Line 5: (A) $count+1$ (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) $n(n-1)/2$
6. Total O: (A) 1 (B) n (C) n^2 (D) $n\log(n)$ (E) n^3

(c) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int i = 1;
2 while (i < n) {
3   i++;
4   if (i > 7) break;
5 }
```

1. Line 1: (A) 1 (B) 0 (C) i (D) n (E) $n+1$
2. Line 2: (A) 8 (B) 7 (C) n (D) $n-1$ (E) $n+1$
3. Lines 3: (A) n (B) $n-1$ (C) 6 (D) 7 (E) 8
4. Lines 4: (A) n (B) $n-1$ (C) 6 (D) 7 (E) 8
5. Tightest Total O: (A) 1 (B) n (C) $\log(n)$ (D) n^2 (E) 2^n

(d) Choose the correct answer:

1. $n^7 + n^4 + n^2 + \log n$ is : (A) $O(n^2)$ (B) $O(n^4)$ (C) $O(n^7)$ (D) $O(\log(n))$ (E) None
2. $2^n + n!$ is : (A) $O(n^2)$ (B) $O(2^n)$ (C) $O(n!)$ (D) $O(n^n)$ (E) None
3. $n + \log n^3 + 6$ is : (A) $O(n)$ (B) $O(\log n^3)$ (C) $O(n \log n)$ (D) $O(n^3)$ (E) None
4. The time complexity of inserting an element in a heap of n elements is:
   (A) $O(n^2)$ (B) $O(n)$ (C) $O(2^n)$ (D) $O(\log(n))$ (E) None
5. $n^2 + n \log n^4$ is : (A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log(n))$ (D) $O(n^4)$ (E) None
6. $n^2 + 1000n$ is : (A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log(n))$ (D) $O(nn^2)$ (E) None
7. $n^4 \log n + n!$ is : (A) $O(n!)$ (B) $O(n^4)$ (C) $O(n^5)$ (D) $O(\log(n))$ (E) None
8. Algorithm A is $O(n)$, and Algorithm B is $O(2n)$. Given the same input:
   (A) A always finishes before B. (B) B always finishes before A. (C) A and B finish at the same time. (D) B requires double the time taken by A. (E) None

Question 1 [30 points]

1. Choose the most appropriate answer:

(1) $n \log(n^3)$ is

(a) $O(\log n)$ (b) $O(n \log n)$ (c) $O(n^3)$ (d) $O(n^3 \log n)$ (e) $O(n)$

```
1 int sum = 0;
2 for (int i = 0; i < n * n; i++)
3         for (int j = n; j < 2 * n; j++)
4                 sum += j;
5 return sum;
```

$n^2 - 0 + 1$ $n^2 + 1$

n^2

n

$2n - n + 1$ $n+1$

$n^3 + n^2$

Choose the correct answer:

Line			Frequency		
1	(a) n	(b) 1	(c) n^2	(d) 0	(e) i
2	(a) n	(b) $n+1$	(c) n^2	(d) $n-1$	(e) n^2+1
3	(a) $i \times n^2$	(b) n^3	(c) n^2+1	(d) n^3+n^2	(e) $n(n+1)/2$
4	(a) n^4	(b) n^3-1	(c) n^2	(d) n^3-n	(e) n^3
5	(a) n	(b) n^3	(c) n^2	(d) 1	(e) n
Total O	(a) $O(n)$	(b) $O(n^2)$	(c) $O(n^3)$	(d) $O(n^4)$	(e) $O(1)$

Q1-1

1- Answer: (b)

2- Answer: (b) O(n log(n)

3- Answer: (e) O(n)

4-Answer: (c) O(1)

5-Answer: (c) O(1) – insert when curren is last, no loop needed.

6-Answer: (c) O(1)

Q1-2

	Statmenet	S/E	Freq	Total	Answer
1	int sum = 0;	1	1	1	b
2	for (int i = 0 ; i < n * n ; i++)	1	n^2+1	n^2+1	e
3	for (int j = n ; j < 2 * n ; j++)	1	$n^2(n+1)$	n^3+n^2	d
4	Sum += j	1	n^3	n^3	e
5	return sum	1	1	1	d
Total O				$O(n^3)$	c

1. Choose the most appropriate answer:

(1) To show that $2n^2 \log n + 2n^3$ is $O(n^3 \log n)$, we can take $c = 4$ and n_0:

(a) -1 (b) -2 (c) 1 (d) O (e) 2

(2) Which of the following is **not** $O(n^2)$

(a) $n^2 \log n$ (b) $2n^2+3$ (c) $n(n+2)/2$ (d) n^2 (e) n

(3) Given an n-element array A of integers, an algorithm searches for the integer '9' and returns true if found. What is the **best**-case running of this algorithm.

a) $O(\log n)$ (b) $O(n \log n)$ (c) $O(1)$ (d) $O(n^2)$ (e) $O(n)$

2. *Consider the following code:*

```
1 System.out.println("glhf");
2 for (int i = 0; i < n * log(n); i++)
3         for (int j = 2; j <= n; j++)
4                 System.out.println("op");
5 System.out.println("gg");
```

Choose the correct answer (**select an answer for each line**):

Line	Frequency				
1	(a) n	(b) -1	(c) 0	(d) 1	(e) $\log n$
2	(a) n	(b) n^2	(c) $n \log n + 1$	(d) $n \log n$	(e) $\log n$
3	(a) n^2	(b) $n^2 \log n$	(c) $n^2 + 1$	(d) $n(n \log n + 1)$	(e) $n(n+1)/2$
4	(a) $n - 1$	(b) n^3	(c) n^2	(d) $n(n \log n)$	(e) $(n-1)n \log n$
5	(a) 0	(b) n	(c) $n \log n$	(d) n^2	(e) 1
Total O	(a) $O(n^2 \log n)$	(b) $O(n^2)$	(c) $O(n \log n)$	(d) $O(1)$	(e) $O(n)$

Question 1 [16 points]

1. Consider the following code:

```
1 for (int i = 0; i < n; i++) {
2         System.out.println("first");
3         for (int j = 0; j <i; j++)
4                 System.out.println("second"); }
5 System.out.println("good␣bye");
```

Choose the correct answer:

Line	Frequency				
1	(a) n	(b) $n+1$	(c) n^2	(d) 0	(e) $n+2$
2	(a) n	(b) $n+1$	(c) n^2	(d) 0	(e) $n-1$
3	(a) n	(b) n^2	(c) $n \log n$	(d) 1	(e) $n(n+1)/2$
4	(a) n	(b) n^2	(c) $n(n-1)/2$	(d) 1	(e) 1
5	(a) n	(b) n^2	(c) 0	(d) 1	(e) $n \log n$
Total O	(a) $O(n)$	(b) $O(n^2)$	(c) $O(n^3)$	(d) $O(n*i)$	(e) $O(1)$

2. Consider the following code:

```
1 int sum = 0;
2 for (int i = 0; i <= n; i++)
3    for (int j = 2; j <n-1; j++)
4          sum += i;
5 return sum;
```

n-1 -2 +1

Choose the correct answer:

Line	Frequency				
1	(a) n	(b) 1	(c) n^2	(d) 0	(e) i
2	(a) n	(b) $n+1$	(c) $n+2$	(d) $n-1$	(e) n^2
~~3~~ 4	(a) $(n+1)(n-3)$	(b) $n(n-2)$	(c) $(n+1)(n-2)$	(d) $n^2(n+1)$	(e) $n(n+1)$
~~4~~ 3	(a) $(n+1)(n-2)$	(b) $n(n-2)$	(c) $(n+1)(n-1)$	(d) $n^2(n+1)$	(e) $n(n+1)$
5	(a) n	(b) n^3	(c) n^2	(d) 1	(e) n
Total O	(a) $O(n)$	(b) $O(n^2)$	(c) $O(n^3)$	(d) $O(n^4)$	(e) $O(1)$

1. For $f(n) = 2n^{\frac{7}{2}} + 6n^2 - n^3$, which of the following is *true*? $f(n)$ is:

$O(n^{\frac{7}{2}})$ (B) $O(n^3)$ (C) $O(n^2)$ (D) A and B (E) None

2. For $f(n) = 14n + 2n\log n - 6n^{\frac{1}{2}}$, which is the *most appropriate*? $f(n)$ is:

(A) $O(n)$ (B) $O(n^{1/2})$ (C) $O(n^2)$ $O(n \log n)$ (E) None

3. For $f(n) = \log(n+1)^2$, which is the *most appropriate*? $f(n)$ is:

$O(\log n)$ (B) $O((\log n)^2)$ (C) $O(\log(n^2))$ (D) $O(n^2)$ (E) None

Suppose you had an algorithm $\mathcal{A}$. When it runs on a list l of size n, $\mathcal{A}$ calls algorithm $\mathcal{B}$ on each of the first three iterations. On the remaining iterations, it calls algorithm $\mathcal{C}$. Suppose $\mathcal{B}$ is $O(n)$ and $\mathcal{C}$ is $O(1)$. Which running time *best describes* algorithm $\mathcal{A}$:

$O(n)$ (B) $O(n^2)$ (C) $O(2^n)$ (D) $O(1)$ (E) None

Suppose that an algorithm $\mathcal{A}$ iterates over a list l of size n. For each element in the first half of l $\mathcal{A}$ executes some $O(\log n)$ operation. For elements in the second half of l, $\mathcal{A}$ executes some $O(1)$ operation. Which running time *best describes* algorithm $\mathcal{A}$?

(A) $O(n)$ $O(n \log n)$ (C) $O(n + \log n)$ (D) $O(n^2 \log n)$ (E) None

For two functions $f(n)$ and $g(n)$, suppose $f(n)$ is $O(g(n))$. What is the best big-O for $f(n) + g(n)$?

$O(g(n))$ (B) $O(2f(n))$ (C) $O(f(n) \cdot g(n))$ (D) $O(f(n)/g(n))$ (E) None

Suppose $f(n)$ is $O(g_1(n))$ and $h(n)$ is $O(g_2(n))$. What is the best big-O of $f(n) \cdot h(n)$?

(A) $O(g_1(n) + g_2(n))$ (B) $O(g_1(n))$ $O(g_1(n) \cdot g_2(n))$ (D) $O(g_2(n))$ (E) None

Choose the correct **frequency** for every line as well as the total O of the following code:

```
1  int sum = 0;
2  for(int i = n; i > 0; i--)
3    for(int j = i; j <= n; j++) {
4      sum = i + j;
5      System.out.println(sum); }
```

1. Line 1: Ⓐ 1 Ⓑ 2 Ⓒ 3 Ⓓ n Ⓔ None
2. Line 2: Ⓐ n Ⓑ $n-1$ Ⓒ $n+1$ Ⓓ $n+2$ Ⓔ None
3. 3 Line 3: Ⓐ n^2 Ⓑ $n(n+3)/2$ Ⓒ $n(n+1)/2$ Ⓓ n^2+n Ⓔ None
4. -3 Line 4: Ⓐ n^2 Ⓑ $n(n-1)/2$ Ⓒ $n(n+1)/2$ Ⓓ $n(n+3)/2$ Ⓔ None
5. -3 Line 5: Ⓐ 1 Ⓑ n^2 Ⓒ $(n+1)/2$ Ⓓ $(2n+1)/2$ Ⓔ None
6. Total O: Ⓐ n^3 Ⓑ n Ⓒ n^2 Ⓓ $n\log(n)$ Ⓔ None

Question 2 .... 8 .... 15 points

Choose the **most appropriate answer** answer:

1. 4 $n\log n + 2^{3\log n} + n$ is : Ⓐ $O(n)$ Ⓑ $O(n^3)$ Ⓒ $O(n\log(n))$ Ⓓ $O(2^n)$ Ⓔ None
2. 4 $n^2 + n\log(n^n)$ is : Ⓐ $O(n^3)$ Ⓑ $O(n^2)$ Ⓒ $O(n\log(n))$ Ⓓ $O(n^2\log(n))$ Ⓔ None
3. $n^2\log 10 + 2000\log n$ is : Ⓐ $O(n^3)$ Ⓑ $O(n\log(n))$ Ⓒ $O(n^2)$ Ⓓ $O(\log(n))$ Ⓔ None
4. $n^{100} + 2^{(n+2)}$ is : Ⓐ $O(n^{100})$ Ⓑ $O(n^n)$ Ⓒ $O(n)$ Ⓓ $O(2^n)$ Ⓔ None
5. Given an n-element array X, Algorithm A chooses $\log n$ elements in X at random and executes an $O(n)$-time calculation if the element is even and $O(n^2)$-time calculation if it is odd. What is the worst-case running time of Algorithm A?:

   Ⓐ $O(n^2)$. Ⓑ $O(n\log n)$. Ⓒ $O(n^2\log n)$. Ⓓ $O(1)$. Ⓔ None

Question 1 ....

Choose the correct frequency for every line as well as the total O of the following code:

```
int sum = 0;
for(int i = n; i > 0; i--) n+1
  for(int j = i; j <= n; j++) {  n-i+2
    sum = i + j;
    System.out.println(sum); }
```

1. Line 1: Ⓐ 1 Ⓑ 2 Ⓒ 0 Ⓓ 3 Ⓔ None
2. Line 2: Ⓐ $i+1$ Ⓑ $n+1$ Ⓒ $n-1$ Ⓓ n Ⓔ None
3. Line 3: Ⓐ $(2n-1)/2$ Ⓑ n^2 Ⓒ $n(n+3)/2$ Ⓓ $n(n+1)/2$ Ⓔ None
4. Line 4: Ⓐ n^2 Ⓑ $n(n+1)$ Ⓒ $n(n-1)/2$ Ⓓ $(2n-1)/2$ Ⓔ None
5. Line 5: Ⓐ $n(n+1)/2$ Ⓑ $(n+1)/2$ Ⓒ n^2 Ⓓ $n(n-1)/2$ Ⓔ None
6. Total O: Ⓐ $n\log(n)$ Ⓑ n^3 Ⓒ n Ⓓ n^2 Ⓔ None

Question 2 ....

Choose the **most appropriate answer** answer:

1. $2^{\log(n)^2} + 2^n$ is : Ⓐ $O(n^2)$ Ⓑ $O(2^n)$ Ⓒ $O(\log n^2)$ Ⓓ $O(4^{\log n})$ Ⓔ Non
2. $n\log(1000^n) + 1000n$ is : Ⓐ $O(n\log(n))$ Ⓑ $O(n)$ Ⓒ $O(\log(1000^n))$ Ⓓ (
3. $n^3 + n\log n^n$ is : Ⓐ $O(n^2\log(n))$ Ⓑ $O(n^n)$ Ⓒ $O(n^2)$ Ⓓ $O(n^3)$ Ⓔ N
4. $n^3/2^{\log(n)} + n\log(n)$ is : Ⓐ $O(n^2)$ Ⓑ $O(n^3)$ Ⓒ $O(n\log(n))$ Ⓓ $O(n^2$ lo
5. For every element of an n-element array X, Algorithm A executes an $O(n\log(n)$ the element is at odd index and $O(n)$-time calculation if it is at even index. Wh running time of Algorithm A?:
   Ⓐ $O(n^2)$. Ⓑ $O(n^3)$. Ⓒ $O(n\log(n))$. Ⓓ $O(n^2\log(n))$. Ⓔ None

...n.

Choose the correct frequency for every line as well as the total O of the following code:

```
Int sum= 0;
for (int i = n ; i > 0 ; i - - )
    for ( int j = i ;  j < = n ; j++) {
        sum= i + j;
        System.out.println(sum); }
```

1. Line 1: (A) 1 (B) 2 (C) 3 (D) n (E) None.
   Answer: A
2. Line 2: (A) n (B) n+1 (C) n-1 (D)n+2 (E) None
   Answer: B
3. Line 3: (A) n^2 (B) n(n-1)/2 (C) (2n+1)/2 (D) n(n+3)/2 (E) None.
   Answer: D
4. Line 4: (A) n^2 (B) n(n-1)/2 (C) (2n+1)/2 (D) n(n+3)/2 (E) None.
   Answer: E
5. Line 5: (A) n^2 (B) n(n-1)/2 (C) n(n+1)/2 (D) (2n+1)/2 (E) None.
   Answer: C
6. Total O: (A) 1 (B) n (C) n^2 (D) n log(n) (E) None.
   Answer: C

Question2:
Choose the **most appropriate answer** answer:

1- n^2 + n log(n^3) is: (A) O(n) (B) O(n^2) (C) O(n log(n)) (D) O(n^4) (E) None
Answer: B

2- n/ log(n) + 1000n is: (A) O(n) (B) O(n^2) (C) O(n log(n)) (D) O(nn^2) (E) None
Answer:

3- n^3 log n + 2^n is: (A) O(2^n) (B) O(n^4) (C) O(n^5) (D) O(log(n)) (E) None
Answer: A

4- log(n^2 + 1) + n is: (A) O(log(n^2)) (B) O(log(n)) (C) O(n^2) (D) O(n) (E) None
Answer:

final Dec-2018

Question 1. . . . . 16 points

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int i = 1;          → 1
2 while (i < n) {     → 8   7
3 i++;                →
4 i f (i > 7) i = n;  1   7
5 }
```

i<n		i>7
i=1	2	2
i=2	3	3
i=3	4	4
i=4	5	5
i=5	6	6
i=6	7	7
i=7	8	8
i=8		

1. Line 1: A.1 B. 0 C .i D. n E. n + 1
2. Line 2: A .1 B. 8 C. n D. n 1 E. n + 1
3. Lines 3 (and similarly 4): A. n B. n 1 C. 6 D.7 E. 8
4. Tightest Total O: A.1 B. n C .log(n) D. n2 . E .2n

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int A = 0;
2 for ( int i = 1; i <= n; i++)
3 for ( int j = i; j < i * 2; j++)
4 A++;
```

1. Line 1: A. 0 B. 1 C. 2 D. n E. A
2. Line 2: A .A B. I C. i + 1 D. n E. n + 1

3. Line 3: A. n2 B. n(n + 1)/2
C. n(n + 1)/2 + 1 D. (n2 + 3n)/2
E. n(n − 1)/2 − 1
4. Line 4: A A2 B n2 C (n2 + 3n)/2
D n2(n + 1)/2 + 1 E n(n + 1)/2
5. Tightest Total O: A .n B. n2 C. n3
D. n4 E .None
(c) Choose the correct answer:
1. n7 + n4 + n2 + log n is : A .O(n2) B. (n4)
C .O(n7) D .O(log(n)) E .None
2. 2n + n! is : A. O(n2) B. O(2n) C. (n!)
D .O( ) E. None
3. n + log n3 + 6 is A .O(n) B. O(log n3)
C. O(n log n) D .O(n3) E. None

Question 1

Choose the correct frequency for every line as well as the total O of the following code:

```
1  int k = 100, sum = 0;
2  for (int i = 0; i < n; i++)
3    for (j = 1; j <= k; j++) {
4      sum = i * j;
5      System.out.println(sum);}
```

1. Line 1: (A) 0 (B) 2 (C) 3 (D) n (E) None
2. Line 2: (A) n (B) $n+1$ (C) $n-1$ (D) $n+2$ (E) None

−3 3. Line 3: (A) $nk-n$ (B) $n(k+2)$ (C) $102n$ (D) $101n$ (E) None

−5 4. Line 4: (A) $100n$ (B) $n(k+2)$ (C) $n(k-1)/2$ (D) n^2 (E) None

−3 5. Line 5: (A) $99n$ (B) $n(k+2)$ (C) $101n$ (D) n^2 (E) None

−5 6. Total O: (A) 1 (B) n (C) nk (D) n^2 (E) None

Question 2 ......12...... 15 points

Choose the **most appropriate answer** answer:

−4 1. $\log(n^2+1)+n$ is : (A) $O(\log n^2)$ (B) $O(\log n)$ (C) $O(n^2)$ (D) $O(n)$ (E) None

2. $n^{100}+2^n$ is : (A) $O(n^{100})$ (B) $O(n)$ (C) $O(n^n)$ (D) $O(2^n)$ (E) None

3. $n^2+\log n^n+n\log n$ is : (A) $O(n)$ (B) $O(n\log(n))$ (C) $O(n^2)$ (D) $O(n^n)$ (E) None

−4 4. $n^3/2^{\log(n)}+n\log(n)$ is : (A) $O(n^2)$ (B) $O(n^3)$ (C) $O(n^2\log(n))$ (D) $O(n\log(n))$ (E) None

5. Given an n-element array X, Algorithm A chooses $n/2$ elements in X at random and executes an $O(n)$-time calculation if the element is even and $O(1)$-time calculation if it is odd. What is the worst-case running time of Algorithm A?:

(A) $O(n)$. (B) $O(1)$. (C) $O(n^2)$. (D) $O(n^3)$. (E) None

16. **for** (**int** i = 0; i < n * n; i ++) {	n^2+1
System .out. println (i);	n^2
for (**int** j = 4; j <= n; j++) {	$n^2(n-1)$
System .out. println (j);	$n^2(n-2)$
}	
}	
System .out. println (" Goodbye !");	1

Choose the correct frequency for every line as well as the total O of the following code:

```
int sum = 0;
for(int i = n; i > 0; i--) n+1
  for(int j = i; j <= n; j++) {  n-i+2
    sum = i + j;
    System.out.println(sum); }
```

1. Line 1: Ⓐ 1 Ⓑ 2 Ⓒ 0 Ⓓ 3 Ⓔ None
2. Line 2: Ⓐ $i+1$ Ⓑ $n+1$ Ⓒ $n-1$ Ⓓ n Ⓔ None
3. Line 3: Ⓐ $(2n-1)/2$ Ⓑ n^2 Ⓒ $n(n+3)/2$ Ⓓ $n(n+1)/2$ Ⓔ None
4. Line 4: Ⓐ n^2 Ⓑ $n(n+1)$ Ⓒ $n(n-1)/2$ Ⓓ $(2n-1)/2$ Ⓔ None
5. Line 5: Ⓐ $n(n+1)/2$ Ⓑ $(n+1)/2$ Ⓒ n^2 Ⓓ $n(n-1)/2$ Ⓔ None
6. Total O: Ⓐ $n\log(n)$ Ⓑ n^3 Ⓒ n Ⓓ n^2 Ⓔ None

Choose the correct frequency for every line as well as the total O of the following code:

```
Int sum= 0;
for (int i = n ; i > 0 ; i - - )
    for ( int j = i ;  j < = n ; j++) {
        sum= i + j;
        System.out.println(sum); }
```

1. Line 1: (A) 1 (B) 2 (C) 3 (D) n (E) None.
   Answer: A
2. Line 2: (A) n (B) n+1 (C) n-1 (D)n+2 (E) None
   Answer: B
3. Line 3: (A) n^2 (B) n(n-1)/2 (C) (2n+1)/2 (D) n(n+3)/2 (E) None.
   Answer: D
4. Line 4: (A) n^2 (B) n(n-1)/2 (C) (2n+1)/2 (D) n(n+3)/2 (E) None.
   Answer: E
5. Line 5: (A) n^2 (B) n(n-1)/2 (C) n(n+1)/2 (D) (2n+1)/2 (E) None.
   Answer: C
6. Total O: (A) 1 (B) n (C) n^2 (D) n log(n) (E) None.
   Answer: C

Question2:

Choose the **most appropriate answer** answer:

1- n^2 + n log(n^3) is: (A) O(n) (B) O(n^2) (C) O(n log(n)) (D) O(n^4) (E) None
Answer: B

2- n/ log(n) + 1000n is: (A) O(n) (B) O(n^2) (C) O(n log(n)) (D) O(nn^2) (E) None
Answer:

3- n^3 log n + 2^n is: (A) O(2^n) (B) O(n^4) (C) O(n^5) (D) O(log(n)) (E) None
Answer: A

4- log(n^2 + 1) + n is: (A) O(log(n^2)) (B) O(log(n)) (C) O(n^2) (D) O(n) (E) None
Answer:

Question3:

Given an n-element array X , Algorithm A chooses n/2 elements in X at random and executes an O(n)-time calculation if the element is even and O(1)-time calculation if it is odd. What is worst-case running time of Algorithm A?:

(A) O(n) (B) O(1) (C) O(n^2) (D) O(n^3) (E) None

(a) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int i = 1;          1
2 while (i < n) {     8      7
3 i++;
4 i f (i > 7) i = n;  1
5 }                   7
```

1. Line 1: A.1 B. 0 C .i D. n E. n + 1
2. Line 2: A .1 B. 8 C. n D. n 1 E. n + 1
3. Lines 3 (and similarly 4): A. n B. n 1 C. 6 D.7 E. 8
4. Tightest Total O: A.1 B. n C .log(n) D. n2 . E .2n

(b) Choose the correct frequency for every line as well as the total O of the following code:

```
1 int A = 0;
2 for ( int i = 1; i <= n; i++)
3 for ( int j = i; j < i * 2; j++)
4 A++;
```

1. Line 1: A. 0 B. 1 C. 2 D. n E. A
2. Line 2: A .A B. I C. i + 1 D. n E. n + 1

3. Line 3: A. n2 B. n(n + 1)/2
C. n(n + 1)/2 + 1 D. (n2 + 3n)/2
E. n(n – 1)/2 – 1
4. Line 4: A A2 B n2 C (n2 + 3n)/2
D n2(n + 1)/2 + 1 E n(n + 1)/2
5. Tightest Total O: A .n B. n2 C. n3
D. n4 E .None
(c) Choose the correct answer:
1. n7 + n4 + n2 + log n is : A .O(n2) B. (n4)
C .O(n7) D .O(log(n)) E .None
2. 2n + n! is : A. O(n2) B. O(2n) C. (n!)
D .O( ) E. None
3. n + log n3 + 6 is : A .O(n) B. O(log n3)
C. O(n log n) D .O(n3) E. None

2.
// fragment #1

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i++)	1	n+1	n+1
b. sum++;	1	n	n
			Total: 2n +1 = O(n)

//fragment #2

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i+=2)	1	ceiling(n/2) +1	ceiling(n/2) +1
b. sum++;	1	ceiling(n/2)	ceiling(n/2)
			Total: **2 (ceiling(n/2))+1 = O(n)**

//fragment #3

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i++)	1	n+1	n+1

b. for(int j = 0; j< n; j++)	1	n(n+1)	n(n+1)
c. sum++;	1	n.n	n^2
			Total:
			$2n^2 + 2n + 1 = O(n^2)$

//fragment #4

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i++)	1	n+1	n+1
b. sum++;	1	n	n
c. for(int j = 0; j< n; j++)	1	n+1	n+1
d. sum++;	1	n	n
			Total:
			$4n + 2 = O(n)$

//fragment #5

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i++)	1	n+1	n+1
b. for(int j = 0; j< n *n; j++)	1	$n(n^2+1)$	$n(n^2+1)$
c. sum++;	1	n. n^2	n^3
			Total:
			$2n^3 + 2n + 1 = O(n^3)$

//fragment #6

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int I = 0; i< n; i++)	1	n+1	n+1
b. for(int j = 0; j< i; j++)	1	$\sum_{i=0}^{n-1} t_i$ where $t_i = i+1$	n(n+1)/2
c. sum++;	1	$\sum_{i=0}^{n-1} (t_i - 1)$ where $t_i = i+1$	n(n-1)/2
			Total:
			$n^2 + n + 1 = O(n^2)$

//fragment #7

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 0; i< n; i++)	1	n+1	n+1
b. . for(int j = 0; j< n*n; j++)	1	$n(n^2+1)$	$n(n^2+1)$
c. for(int k = 0; k< j; k++)	1	$n \sum_{j=0}^{n^2-1} t_j$ where $t_j = j+1$	$n^3 (n^2+1)/2$
d. sum++;	1	$n \sum_{j=0}^{n^2-1} (t_j -1)$ where $t_j = j+1$	$n^3 (n^2-1)/2$
			Total: $\mathbf{n^5 + n^3 + 2n+1 = O(n^5)}$

//fragment #8

Statement	Steps per Execution	Times/ Frequency	Total Steps
a. for(int i = 1; i< n; i= i*2)	1	ceiling(log n) +1	ceiling(log n) +1
b. sum++;	1	ceiling(log n)	ceiling(log n)
			Total: 2 (ceiling(log n))+1 = O(log n)