King Saud University
College of Computer and Information Sciences
Computer Science Department

CSC 212 **Second Semester 1439-1440**

Tutorial #2

Question 1: Find the total number of primitive operations and the Big Oh notation of the following methods:

a)

	Statements	**S/E**	**Freq.**	**Total**
1	`void findProduct(int n)`	**0**	**0**	**0**
2	`{`	**0**	**0**	**0**
3	`    int product = 1;`	**1**	**1**	**1**
4	`    for (int i = 0; i < 10; i = i + 2)`	**1**	**6**	**6**
5	`    {`	**0**	**0**	**0**
6	`      product = product * i;`	**1**	**5**	**5**
7	`    }`	**0**	**0**	**0**
8	`    System.out.println(product);`	**1**	**1**	**1**
9	`}`	**0**	**0**	**0**
	Total Operations	**13**		
	Big Oh	**O(1)**		

b)

	Statements	S/E	Freq.	Total
1	`void findNestedProduct(int n)`	0	0	0
2	`{`	0	0	0
3	`int product = 1;`	1	1	1
4	`for (int i = 0; i < n; i = i + 2)`	1	(n/2) + 1	(n/2) + 1
5	`{`	0	0	0
6	`for (int j = 0; j < 6; j ++)`	1	7(n/2)	7(n/2)
7	`product = product * i * j;`	1	6(n/2)	6(n/2)
8	`}`	0	0	0
9	`System.out.println(product);`	1	1	1
10	`}`	0	0	0
	Total Operations	7n + 3		
	Big Oh	O(n)		

c)

	Statements	S/E	Freq.	Total
1	`void calcSum(int n)`	0	0	0
2	`{`	0	0	0
3	`int sum = 0;`	1	1	1
4	`for (int i = 1; i <= n; i++)`	1	n+1	n+1
5	`{`	0	0	0
6	`for (int j = 1; j <= i; j++)`	1	$\frac{n(n+1)}{2}+n$	$\frac{n(n+1)}{2}+n$
7	`sum = sum + 1;`	1	$\frac{n(n+1)}{2}$	$\frac{n(n+1)}{2}$
8	`}`	0	0	0
9	`System.out.println(sum);`	1	1	1
10	`}`	0	0	0
	Total Operations	n^2+3n+3		
	Big Oh	$O(n^2)$		

Note:

Frequency of line 6: $\sum_{i=1}^{n} i + 1$

Frequency of line 7: $\sum_{i=1}^{n} i$

Is there any other way to solve the above problem in better performance?

Yes:

1) **In O(n)**

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

2) **In O(1)**

```
return n * (n + 1) / 2
```

d)

	Statements	**S/E**	**Freq.**	**Total**
1	`public int method4(int n)`	**0**	**0**	**0**
2	`{`	**0**	**0**	**0**
3	`int sum = 0;`	**1**	**1**	**1**
4	`for (int i = 0; i < n; i++)`	**1**	**n + 1**	**n + 1**
5	`   for (int j = n; j >= 1; j = j/2) {`	**1**	**n(logn + 2)**	**n(logn + 2)**
6	`     sum += 1;`	**1**	**n(logn + 1)**	**n(logn + 1)**
7	`   }`	**0**	**0**	**0**
8	`return sum;`	**1**	**1**	**1**
9	`}`	**0**	**0**	**0**
	Total Operations	**2nlogn + 4n + 3**		
	Big Oh	**O(nlogn)**		