Order Analysis and Summations – Practice Problems

Order Analysis Problems

1. For an O(n!) algorithm, a problem instance of size n = 8 requires 60 seconds to solve. How long will it take to solve a problem instance of size n = 7?

2. For an $O(2^n)$ algorithm, a problem instance of size n = 7 requires 96 seconds to solve. If you used a different sized problem instance and the algorithm required 12 seconds to solve the problem, what was the size of this problem instance?

3. For an $O(n / \log_2 n)$ algorithm, a problem instance of size n = 16 requires 96 milliseconds to solve. How long would it take the algorithm to solve a problem instance of size n = 8?

4. For an $O(n \log_2 n)$ algorithm, a problem instance of size n = 16 requires 32 seconds to solve. How long will it take to solve a problem instance of size n = 8?

Loop Analysis

1. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine what the final value of *x* will be in terms of n.

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

(a)

(b)

2. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = n; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                x = x + j;
```

(a)

(b)

3. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                if (j < i)
                        x = x + 1;
```

(a)

(b)

4. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n–2; j++)
                x = x + j;
```

(a)

(b)

5. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                if (j == i)
                        x = x + 1;
```

(a)

6. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

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

(a)

(b)

7. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = n*(8*n+8); i++)
        for (j = 1; j <= n; j++)
                x = x + (n – j);
```

(a)

(b)

Solutions

Order Analysis Problems

5. For an O(n!) algorithm, a problem instance of size n = 8 requires 60 seconds to solve. How long will it take to solve a problem instance of size n = 7?

$$\frac{8!}{60} = \frac{7!}{t} \Rightarrow (8!)t = 60(7!) \Rightarrow t = \frac{60(5040)}{8!} = \frac{302400}{40320} = 7.5 \text{ sec}$$

6. For an O(2^n) algorithm, a problem instance of size n = 7 requires 96 seconds to solve. If you used a different sized problem instance and the algorithm required 12 seconds to solve the problem, what was the size of this problem instance?

$$\frac{2^7}{96} = \frac{2^n}{12} \Rightarrow 2^n = \frac{(2^7)12}{96} = 16 \Rightarrow \log_2(16) = n \Rightarrow n = 4$$

7. For an O(n / $\log_2$n) algorithm, a problem instance of size n = 16 requires 96 milliseconds to solve. How long would it take the algorithm to solve a problem instance of size n = 8?

$$\frac{16/\log_2 16}{96} = \frac{8/\log_2 8}{t} \Rightarrow \frac{16/4}{96} = \frac{8/3}{t} \Rightarrow \frac{16}{96(4)} = \frac{8}{3t}$$

$$\Rightarrow 48t = 3072 \Rightarrow t = 64 \text{ msec}$$

8. For an O(n $\log_2$ n) algorithm, a problem instance of size n = 16 requires 32 seconds to solve. How long will it take to solve a problem instance of size n = 8?

$$\frac{16(\log_2 16)}{32} = \frac{8(\log_2 8)}{t} \Rightarrow \frac{16(4)}{32} = \frac{8(3)}{t} \Rightarrow 64t = 32(24) \Rightarrow t = \frac{768}{64} = 12 \text{ sec}$$

Loop Analysis

8. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine what the final value of *x* will be in terms of n.

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

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{2n} 1 = 2n(2n) = 4n^2 \quad \Rightarrow \quad O(n^2)$

(b) $\sum_{i=1}^{2n}\sum_{j=1}^{i} 3 = 3\sum_{i=1}^{2n}\sum_{j=1}^{i} 1 = 3\sum_{i=1}^{2n} i = 3\frac{2n(2n+1)}{2} = 3(n(2n+1)) = 6n^2 + 3n$

9. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = n; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                x = x + j;
```

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{n} 1 - \sum_{i=1}^{n-1} 1 = (2n)(n) - \frac{(n-1)(n)}{2} = 2n^2 - \frac{n^2 - n}{2} = \frac{4n^2 - n^2 - n}{2} = \frac{3n^2 - n}{2} = O(n^2)$

(b) $\sum_{i=n}^{2n}\sum_{j=1}^{n} j = \sum_{i=1}^{2n}\sum_{j=1}^{n} j - \sum_{i=1}^{n-1}\sum_{j=1}^{n} j = 2n\frac{n(n+1)}{2} - (n-1)\frac{n(n+1)}{2} = \frac{4n(n^2+n)}{2} - \frac{(2n-2)(n^2+n)}{2}$

$$= \frac{4n^3 + 4n^2}{2} - \frac{2n^3 + 2n^2 - 2n^2 - 2n}{2} = \frac{4n^3 + 4n^2 - 2n^3 - 2n^2 + 2n^2 + 2n}{2}$$

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

10. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                if (j < i)
                        x = x + 1;
```

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{n} 1 = \sum_{i=1}^{2n} n = n\sum_{i=1}^{2n} 1 = n(2n) = O(n^2)$

(b) $\sum_{i=1}^{2n}\sum_{j=1}^{n} 1 - \sum_{j=1}^{n} j = n\sum_{i=1}^{2n} 1 - \frac{n(n+1)}{2} = n(2n) - \frac{n(n+1)}{2} = 2n^2 - \frac{n(n+1)}{2} = \frac{4n^2 - n^2 - n}{2} = \frac{3n^2 - n}{2}$

11. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n–2; j++)
                x = x + j;
```

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{n-2} 1 = 2n(n-2) = 2n^2 - 4n = O(n^2)$

(b) $x = \sum_{i=1}^{2n}\sum_{j=1}^{n-2} j = \sum_{i=1}^{2n} \frac{(n-2)(n-1)}{2} = \frac{(n-2)(n-1)}{2}\sum_{i=1}^{2n} 1 = 2n\frac{(n-2)(n-1)}{2} = n(n-1)(n-2)$

12. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = (2*n); i++)
        for (j = 1; j <= n; j++)
                if (j == i)
                        x = x + 1;
```

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{n} 1 = \sum_{i=1}^{2n} n = n(2n) = 2n^2 = O(n^2)$

(b) Notice that *j* is only equal to *i* once for each value of *i*. Since *j* only goes to the limit of *n*, the statement x=x+j is only executed *n* times, thus:

$$x = \sum_{j=1}^{n} j = \frac{n(n+1)}{2}$$

13. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

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

(a) $\sum_{i=1}^{2n}\sum_{j=1}^{3n} 1 = (2n)(3n) = 6n^2 = O(n^2)$

(b) $x = \sum_{i=1}^{2n}\sum_{j=1}^{3n} i = 3n\sum_{i=1}^{2n} i = 3n\frac{2n(2n+1)}{2} = 3n\frac{4n^2+2n}{2} = \frac{12n^3+6n^2}{2} = 6n^3+3n^2$

14. For the code segment shown below, (a) find the Big-Oh order of this code segment and (b) determine the final value of x in terms of n.

```
x = 0;
for (i = 1; i < = n*(8*n+8); i++)
        for (j = 1; j <= n; j++)
                x = x + (n – j);
```

(a) $\sum_{i=1}^{8n^2+8n} \sum_{j=1}^{n} 1 = n(8n^2 + 8n) = 8n^3 + 8n^2 = O(n^3)$

(b) $x = \sum_{i=1}^{8n^2+8n} \sum_{j=1}^{n-1} j = \sum_{i=1}^{8n^2+8n} \left(\frac{n^2 - n}{2} \right) = (8n^2 + 8n)\left(\frac{n^2 - n}{2} \right) = 4n^4 - 4n^2$