## 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 | $n \log n \, (n - 2 + 1)$ | $n^2 \log n - n \log n$ |
| 5 | System.out.println(j); | 1 | $n \log n \, (n - 2)$ | $n^2 \log n - 2n \log n$ |
| 6 | } | 0 | - | - |
| 7 | System.out.println("Goodbye!"); | 1 | 1 | 1 |
| 8 | } | 0 | - | - |
| **Total** | | $2n^2 \log n - n \log n + 2$ | | |
| **Big Oh** | | $O(n^2 \log n)$ | | |

| | 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** | | $O(n^3)$ | | |

| | Statements | S/E | Freq. | Total |
|---|---|---|---|---|
| 1 | public void func3(int n) { | 0 | - | - |
| 2 | for (int i = n; i > 0; i--) { | 1 | $n+1$ | $n+1$ |
| 3 | System.out.println(i); | 1 | $n$ | $n$ |
| 4 | for (int j = 0; j < i; j++) | 1 | $\frac{(n+1)\,(n+2)}{2}-1$ | $\frac{n^2+3n}{2}$ |
| 5 | System.out.println(j); | 1 | $\frac{n(n+1)}{2}$ | $\frac{n^2+n}{2}$ |
| 6 | } | 0 | - | - |
| 7 | System.out.println("Goodbye!"); | 1 | 1 | 1 |
| 8 | } | 0 | - | - |
| **Total** | | $n^2+4n+2$ | | |
| **Big Oh** | | $O(n^2)$ | | |

- In order to find the frequency of line number 4 and 5, we will trace them assuming ***n = 3***:

| i | j | Loop Checking Step | Loop Internal Step |
|---|---|---|---|
| 3 | 0 | Executed 4 times | Executed 3 times |
| | 1 | | |
| | 2 | | |
| | 3 | | |
| 2 | 0 | Executed 3 times | Executed 2 times |
| | 1 | | |
| | 2 | | |
| 1 | 0 | Executed 2 times | Executed 1 times |
| | 1 | | |

- **Line 4:** from above tracing, the minimum number of checking is ***2*** and the maximum is 4 which is ***n+1***. By applying the summation formula:

$$\sum_{i=min}^{max} i = \frac{\max\,(max+1)}{2} - \frac{\min\,(min-1)}{2} = \sum_{i=2}^{n+1} i = \frac{(n+1)\,(n+2)}{2} - \frac{2\,(2-1)}{2}$$

$$\sum_{i=2}^{n+1} i = \frac{n^2+3n+2}{2} - 1 = \frac{n^2+3n}{2}$$

- **Line 5:** from above tracing, the minimum number of loop internal steps execution is ***1*** and the maximum is 3 which is ***n***. By applying the summation formula:

$$\sum_{i=min}^{max} i = \frac{\max\,(max+1)}{2} - \frac{\min\,(min-1)}{2} = \sum_{i=1}^{n} i = \frac{\mathrm{n}(\mathrm{n}+1)}{2} - \frac{1(1-1)}{2}$$

$$\sum_{i=1}^{n} i = \frac{\mathrm{n}^2 + n}{2}$$

- **IMPORTANT: there is no need to multiply the summation result by *n* (from the outer loop) since summation formula takes that into consideration.**

| | Statements | S/E | Freq. | Total |
|---|---|---|---|---|
| 1 | public void func4(int n) { | 0 | - | - |
| 2 | int m = 1; | 1 | 1 | 1 |
| 3 | while ( m <= n ) { | 1 | $n-1+1+1$ | $n+1$ |
| 4 | system . out . println (m); | 1 | $n$ | $n$ |
| 5 | i = n; | 1 | $n$ | $n$ |
| 6 | while (i > 0 ) { | 1 | $n(\lfloor \log n \rfloor + 1 + 1)$ | $n\lfloor \log n \rfloor + 2n$ |
| 7 | system .out . println (i); | 1 | $n(\lfloor \log n \rfloor + 1)$ | $n\lfloor \log n \rfloor + n$ |
| 8 | i = i / 2; | 1 | $n(\lfloor \log n \rfloor + 1)$ | $n\lfloor \log n \rfloor + n$ |
| 9 | } | 0 | - | - |
| 10 | m++; | 1 | $n$ | $n$ |
| 11 | } | 0 | - | - |
| 12 | } | 0 | - | - |
| **Total** | | $3n \log n + 8n + 2$ | | |
| **Big Oh** | | $O(n \log n)$ | | |