Practical Exercise 9.1 – Algorithms Analysis and Big-O Notation

Overall Objective

To estimate algorithm efficiency using the Big-O notation and determine the complexity of various types of algorithms.

Description

1. Why is a constant factor ignored in the Big-O notation? *[refer to Ch24 slide 7]*

2. Why is a non-dominating term ignored in the Big-O notation? *[refer to Ch24 slide 8]*

3. What is the order of each of the following functions?*[refer to Additional Note slide 21,22]*

   a. $\frac{(n^2+1)^2}{n}$

   b. $\frac{(n^2+\log_2 n)^2}{n}$

   c. n^3+100n^2+n

   d. 2^n+100n^2+45n

   e. $n2^n+n^2 2^n$

4. Use the Big-O notation to estimate the time complexity of the following methods:

   a.
```
public static void method1(int n) {
     for(int i = 0; i < n; i += 2) {
         System.out.print(Math.random() + " ");
     }
}
```
   [refer to Ch24 slide 11 and Additional Note slide 13, 21, 22]

   b.
```
public static void method2(int n) {
     for(int i = 0; i <= n; i++) {
         for(int j = 0; j < i; j++) {
             System.out.print(Math.random() + " ");
         }
     }
}
```
   [refer to Ch24 slide 13 and Additional Note slide 19, 21, 22]

   c.
```
public static void method3(int[] m) {
     for(int i = 0; i < m.length; i += 2) {
         System.out.print(m[i] + " ");
     }

     for(int i = m.length - 1; i >= 0; i -= 2) {
         System.out.print(m[i] + " ");
     }
}
```
   [refer to Ch24 slide 15]

5. Put the following growth functions in order:
   [refer to Ch24 slide 30]

   $$\frac{5n^3}{4032},\quad 44\log n,\quad 500,\quad \frac{2^n}{45},\quad 10n\log n,\quad 2n^2,\quad 3n$$

6. Design/describe an algorithm for the following tasks, and analyse the time complexity of the algorithm.

   a. Compute the sum of all numbers from n1 to n2 for (n1 < n2).

   b. Find the occurrence of the largest element in an array.

   c. Remove duplicate element in an array.

7. What is dynamic programming? Give an example. (refer to recursive and non-recursive Fibonacci solutions) *[refer to Ch24 slide 36]*