CHAPTER 1 – PROBLEMS

Problem 1. For each of the following algorithms, indicate (i) a natural size metric for its inputs, (ii) its basic operation, and (iii) whether the basic operation count can be different for inputs of the same size:

a. computing the sum of n numbers
b. computing $n!$
c. finding the largest element in a list of n numbers
d. Euclid's algorithm to find the GCD of two integers.

Problem 2. Calculate the number of all assignment and comparison operations of the following algorithms, then show their order of growth in term of O-notation:

```
a. for (i = 0; i < n; i++)
      for (j = 0; j < n; j++)
        b[i][j] += c;
```

```
b. for (i = 0; i < n; i++)
      if (a[i] == k)
          return 1;
      return 0;
```

```
c. for (i = 0; i < n; i++)
      for (j = i+1; j < n; j++)
          b[i][j] -= c;
```

Problem 3. For each of the following pairs of functions, indicate whether the first function of each of the following pairs has a lower, same, or higher order of growth (to within a constant multiple) than the second function.

a. $n(n+1)$ and $2000n^2$
b. $100n^2$ and $0.01n^3$
c. $\log_2 n$ and $\ln n$
d. ${\log^2}_2 n$ and $\log_2 n^2$
e. $2^{n-1}$and 2^n
f. $(n-1)!$ and $n!$

Problem 4. List the following functions according to their order of growth from the lowest to the highest:

$$(n-2)!, 5\lg(n+100)^{10}, 2^{2n}, 0.001n^4+3n^3+1, \ln^2 n, \sqrt[3]{n}, 3^n$$

Problem 5. Express the function $\frac{n^3}{1000}-100n^2-100n+3$ in terms of O-notation.

Problem 6. Explain why the statement, “The running time of algorithm A is at least $O(n^2)$,” is meaningless.

Problem 7. Show that :

a. $f(x) = 4x^2 - 5x + 3 \in O(x^2)$
b. $f(x) = (x + 5)\, log_2(3x^2 + 7) \in O(x\, log_2 x)$
c. $f(x) = (x^2 + 5\, log_2 x)/(2x + 1) \in O(x)$

Problem 8. Are the following functions $O(x)$?

a. $f(x) = 10$
b. $f(x) = 3x + 7$
c. $f(x) = 2x^2 + 2$

Problem 8. Describe the running time of the following function using O-notation:

$$S = 1 + \frac{1}{2} + \frac{1}{6} + \cdots + \frac{1}{n!}$$

Problem 9. Find $g(n)$ of the following $f(n)$ so that $f(n) \in O(g(n))$.

a. $f(n) = (2 + n) * (3 + \log_2 n)$
b. $f(n) = 11 * \log_2 n + \frac{n}{2} - 3542$
c. $f(n) = n * (3 + n) - 7n$
d. $f(n) = \log_2(n^2) + n$

Problem 10. Determine $O(g(n))$ of the following functions:

a. $f(n) = 10$
b. $f(n) = 5n + 3$
c. $f(n) = 10n^2 - 3n + 20$
d. $f(n) = \log n + 100$
e. $f(n) = n\, log\, n + log\, n + 5$

Problem 11. Which one is correct? Explain your answer.

a. $2^{n+1} = O(2^n)$?
b. $2^{2n} = O(2^n)$?

Problem 12. Write the algorithms to solve the following problems using C++ and recursion. Find the Big-O of your algorithms.

a. Find the maximum of an array of *n* integers.
b. Calculate the factorial of an integer *n.*
c. Calculate the sum of *n* integers.
d. Check if an array of *n* elements is symmetric or not.
(Example of a symmetric array: $< 12, 4, 3, 4, 12 >, < 5, 19, 19, 5 >$)