



# Computer Science 431 Algorithms

The College of Saint Rose  
Spring 2015

## Problem Set 2: Asymptotic Analysis

Due: 11:59 PM, Friday, February 6, 2015

You may work alone or in groups of size 2 or 3 on this assignment. However, in order to make sure you learn the material and are well-prepared for the exams, you should work through the problems on your own before discussing them with your group members, should you choose to work in a group. Only one submission per group is needed.

There is a significant amount of work to be done here, and you are sure to have questions. It will be difficult if not impossible to complete the problem set if you wait until the last minute. A slow and steady approach will be much more effective.

---

### Getting Set Up

Create a directory/folder for your work on this problem set. As you prepare your solution to the written problems, you are encouraged, but not required to use  $\LaTeX$ . Ultimately, be sure your solutions are in a single PDF file. Place all code for programming tasks in this folder.

---

### Programming Task

Sometimes an exhaustive search algorithm considers all permutations of a set. It is not obvious though how to generate these permutations in a program. Write a program that prints all permutations of integers  $1..n$ . Your program should take  $n$  as an input value, and it should work for any  $n \geq 1$ . You may write your program in Java, C, or C++. If you would like to use a different language, ask, and it will likely be approved. (10 points)

---

### Problem Set Problems

#### ? Question 1:

| Levitin Exercise 2.3.1, p. 67 (4 points)

#### ? Question 2:

| Levitin Exercise 2.3.2, p. 67 (2 points)

#### ? Question 3:

| Compare the asymptotic growth rates of the following pairs of functions and decide if one grows faster than the other or if they grow at the same rate. Prove your answers using limits. All logarithms are base 2, unless otherwise noted. (5 points)

1.  $5n^2 + 2n + 5$  and  $n^2$
2.  $n \log n$  and  $n^2$
3.  $2^{\log n}$  and  $n$
4.  $2^n$  and  $2^{2n}$
5.  $n^r$  and  $n^s$ , where  $r$  and  $s$  are arbitrary fixed constants such that  $r > s > 0$ .

### ? Question 4:

Prove the following using limits (4 points):

1.  $n \in \Omega(\log n)$
2.  $5n + 6 \in \Theta(n)$
3.  $n \log n \in \Omega(n)$
4.  $n \in \Theta(n - n^{\frac{1}{2}})$

### ? Question 5:

For each of the following functions, indicate the class  $O(g(n))$  to which the function belongs. Use the “smallest”  $g(n)$  possible to obtain the tightest bound. You need not prove your assertions in this case, but justify your choice informally. (2 points)

1.  $\frac{1}{n} + 12$
2.  $\frac{\sin n}{n}$

### ? Question 6:

For each of the following functions, indicate the class  $O(g(n))$  to which the function belongs. Use the “smallest”  $g(n)$  possible to obtain the tightest bound, unless otherwise specified. Prove your assertions using the definition of  $O(g(n))$  (i.e., produce constants). (6 points)

1.  $762n^2 - 56n + 37$
2.  $(n^3 + 100)^5$
3.  $n \log n$ , use  $g(n) = n^2$

**? Question 7:**

For each of the following functions, indicate the class  $\Omega(g(n))$  to which the function belongs. Use the “largest”  $g(n)$  possible to obtain the tightest bound unless otherwise specified. Prove your assertions using the definition of  $\Omega(g(n))$  (*i.e.*, produce constants). (6 points)

1.  $762n^2 - 56n + 37$
2.  $2^n + n^2$
3.  $n \log n$ , use  $g(n) = n$

**? Question 8:**

For each of the following functions, indicate the class  $\Theta(g(n))$  to which the function belongs. Prove your assertions using the definition of  $\Theta(g(n))$  (*i.e.*, produce constants). (5 points, 1 for the first, 4 for the second)

1.  $762n^2 - 56n + 37$  (hint: you’ve already done this, just bring it all together)
2.  $n^4 + 50n^2 - 23$

**? Question 9:**

Analyze the following code segment and give an exact formula involving summations for the number of times the word “hello” is printed. Then simplify your formula to obtain an exact formula for the number of times the word “hello” is printed that does not involve summations. (In the code segment, `Math.pow(2, j)` is a function that returns 2 raised to the  $j^{\text{th}}$  power.) (6 points)

```
for ( i = 1; i <= n; i++ )
    for ( j = 1; j <= n; j++ ) {
        for ( k = 1; k <= Math.pow(2, j); k++ ) {
            System.out.println( "hello" );
        }
        System.out.println( "hello" );
    }
```

Note: you may test your formula by implementing this code and then running it for different values of  $n$ .

**Submitting**

Before 11:59 PM, Friday, February 6, 2015, submit your problem set for grading. To complete the submission, upload an archive (a .7z or .zip) file containing all required files using Submission Box at <http://sb.teresco.org> under assignment "PS2".

---

## Grading

This assignment is worth 50 points, which are distributed as follows:

Feature	Value	Score
Permutation Generator Program	10	
Question 1	4	
Question 2	2	
Question 3	5	
Question 4	4	
Question 5	2	
Question 6	6	
Question 7	6	
Question 8	5	
Question 9	6	
Total	50	