CISC 380 **Instructor: Dr. Miracle**

Homework 1

Student: Haoyuan Du, Josh Ringling

Purpose:

- Improve your understanding of O, Θ, Ω, o, and ω notation.
- Practice designing and analyzing algorithms (and writing pseudocode).

General Homework Policies:

- This homework is due by the start of class **on Monday, 2/17**. In order to submit your homework, you need to do two things.
  1. Submit the `pdf` on Canvas (it must be generated by modifying the LaTeX template provided).
  2. Bring a paper print-out of your assignment to class.
- You will be assigned a partner on this assignment. Partner assignments will be posted to Canvas in a separate file. Only one assignment should be submitted and you and your partner will both receive the same grade. Make sure to include both you and your partner's name on the homework.
- If you choose to work with a partner, your assignment should be a true joint effort, equally created, and understood by both partners.
- Getting *ANY* solutions from the web, previous students etc. is *NOT* allowed.[1]
- You are not allowed to discuss this assignment with anyone except for your partner (if you have one) or the instructor.[1]
- Your work will be graded on correctness and clarity. Write complete and precise answers and show all of your work (there is a detailed grading rubric included at the end of the assignment).
- Questions marked (*PRACTICE*) will not be graded and do not need to be submitted. However it's highly recommended that you complete them.

[1]*See the section of the syllabus on academic dishonesty for more details.*

1. (4 points) For each pair of functions below, list which one(s) of the following are true: (1) $f(n) = o(g(n))$, (2) $f(n) = O(g(n))$, (3) $f(n) = \Theta(g(n))$, (4) $f(n) = \Omega(g(n))$, or (5) $f(n) = \omega(g(n))$.

   (a) $f(n) = 2^n$, $g(n) = 3^n$.

   Solution: (1)(2)

   (b) $f(n) = \log(n + 5)$, $g(n) = \log(25n^2)$.

   Solution: (2)(3)(4)

   (c) $f(n) = \ln(n)$, $g(n) = \log_5(n)$.

> **Solution:** (2)(3)(4)

(d) $f(n) = n \log n + \ln^6 n$, $g(n) = n^{1/3} \log^2 n$.

> **Solution:** (4)(5)

(e) *(PRACTICE)* $f(n) = 2^{n/2}$, $g(n) = 2^{n+2}$.

2. (2 points) Suppose $T_1(n) = O(f(n))$ and $T_2(n) = O(f(n))$. Which of the following are true? Explain why or give a counterexample.

   (a) $\frac{T_1(n)}{T_2(n)} = O(1)$

   > **Solution:** False. Counterexample as below.
   > Suppose $f(n) = n^3$, $T_1(n) = n^2$ and $T_2(n) = n$. $T_1(n) = O(n^3)$, $T_2(n) = O(n^3)$. But $\frac{T_1(n)}{T_2(n)} = n$ instead of $O(1)$. So it's false.

   (b) *(PRACTICE)* $T_1(n) - T_2(n) = O(f(n))$

   (c) *(PRACTICE)* $T_1(n) - T_2(n) = o(f(n))$

   (d) *(PRACTICE)* $T_1(n) = O(T_2(n))$

3. (2 points) Suppose we have three functions $f(n), g(n)$, and $h(n)$ such that $f(n) = O(g(n))$ and $g(n) = \Omega(h(n))$. Does this imply that $f(n) = \Omega(h(n))$? Explain why or give a counterexample.

> **Solution:** False. Counterexample as below.
> Let $f(n) = n^2, g(n) = n^4, h(n) = n^3$. Then $f(n) = O(n^4)$ and $g(n) = \Omega(n^3)$. However, $n^2 \neq \Omega(n^3)$, so $f(n) \neq \Omega(h(n))$. Thus, it's false.

4. (3 points) For each of the following program fragments give a Θ bound on the asymptotic running time. Show your work and justify your answer (this should include justifying why your answer is a lower AND an upper bound).

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

   > **Solution:** The value of i will be doubled each time when the inner loop finished each time, such as: i = 1, 2, 4, 8..., $\frac{n}{2}$, n. There are $\log n$ terms in total. So we can get $\sum_{i=0}^{\log n} 2^i$ which equals $2^{\log_2^{n-1}}$ => n-1. Thus, the θ bound is $\theta(n)$

   (b) *(PRACTICE)*
```
Sum = 0;
for (i=0; i< n; i++)
   for(j=0; j < i; j++)
```

```
        for (k=0; k<5; k++)
           Sum = Sum + 1;
```

(c) (*PRACTICE*)

```
Sum = 0;
for (i=0; i< n; i++)
   for(j=5; j < n*n; j++)
      Sum = Sum + 1;
```

5. (5 points) For this problem you will write an algorithm to determine how many students are taking both CISC 380 and CISC 340 this term. Assume that the class lists are stored in two arrays each consisting of unique student IDs in arbitrary order. To keep it simple, we assume that the two classes have the same number of students, denoted by n. *Faster (in Θ notation) and correct will receive more credit.*

   (a) Write an algorithm in pseudocode to count the number of students who are taking both courses this term. Include a brief (several sentences) explanation of how your algorithm works **BEFORE** the pseudocode and don't forget the pseudocode guidelines we discussed in class!

Solution: Firstly, we use merge sort to sort CISC380 array in order. Secondly, we take each element from CISC340 (unordered array) as a target and use binary search in CISC380 array (ordered) because binary search must be used in ordered arrays. If it found the same element, the counter will be increased by 1.

Pseudocode:

```
function FindSameStudents(CISC380[$a_1, ...a_n$], CISC340[$a_1, ...a_n$])
    count = 0
    sorted380 = MergeSort(CISC380[$a_1, ...a_n$])
    for i=0 to n, i++ do:
        if BinarySearch(sorted380[$a_1, ...a_n$],CISC340[i])
            count++
    return count
```

   (b) Give a big-O bound on the asymptotic running time (tight bounds for full credit). Show your work and justify your answer.

Solution: Assumed there are n elements in both arrays.
Firstly, the MergeSort function takes $\theta(nlogn)$ to sort the CISC380. Secondly, we use a for loop which has n times. Inside the loop, the BinarySearch function takes $\theta(logn)$, and the comparison takes constant time. So the loop in total takes $n * (logn + c)$ time. Thus, for the whole pseudocode, it takes $2 * nlogn + n * c$ which is $\theta(nlogn)$.

Grading Criteria:

Your work will be graded on both correctness **and clarity**. Write complete and precise answers and show all of your work. Your pseudo-code and proofs should follow the guidelines posted on

Canvas and discussed in class.

Evaluation Rubric for Problems 1(a) - (d).

Component	Requirement for Full Credit
Solution Correctness (1pt)	The answer is correct.

Evaluation Rubric for Problems 2-3.

Component	Requirement for Full Credit
Solution Correctness (1pt)	The answer is correct.
Written Explanation (1pt)	Your solution includes a logical argument explaining why your answer is correct. The explanation is clear, correct and complete. It includes full sentences and is easily understood by any student in the class. Think of this as a proof and remember what you learned in Math 128.

Evaluation Rubric for Problems 4.

Component	Requirement for Full Credit
Solution Correctness (1pt)	The answer is correct.
Written Explanation (2pt)	Your solution includes a logical argument explaining why your answer is correct. The explanation is clear, correct and complete. It includes full sentences and is easily understood by any student in the class. Think of this as a proof and remember what you learned in Math 128. This should include justifying why your answer is a lower AND an upper bound.

Evaluation Rubric for Problem 5.

Component	Requirement for Full Credit
Solution Correctness (2pts)	The algorithm will work correctly on all inputs. The running time is the fastest (in terms of O-notation) that can be obtained using the tools given.
Written Description (1pt)	The explanation is concise and captions the behavior of the algorithm in the general case. It includes full sentences and should be easily understood by any student in the class. It shouldn't require seeing the pseudo-code to understand the explanation.
Pseudo-code (1pt)	The pseudo-code is clearly written following the guidelines discussed in class.
Running Time Analysis (1pt)	The solution gives both a bound and an explanation. Both are clearly written and correct (for the algorithm given!).