CMPS 340 — Design and Analysis of Algorithms

Assignment 1

Due: *Saturday, September 19, 2020 by 11:50 pm*
Instructor: Aminul Islam, `aminul@louisiana.edu`
Grader: Tingting Yang, `c00304740@louisiana.edu`
Grader's Office hour: Mon. & Wed. 10:00 am - 12:00 pm on Zoom
Grader's Zoom meeting link: `https://ullafayette.zoom.us/j/99532884029`

- This is an individual assignment.
- The assignment can be turned in only until 11:50 pm on Sunday (9/20/2020) with 20% penalty on the assignment points.
- The complete name and ULID of students must be written within the document.
- Each student should submit:
  - A document that includes answer of Questions 1, 2, 3, 4, 5 and source code of Question 6.
  - Bug-free, clean, and easily executable source code (compulsory for question 6).
  - The code can be written in any of the imperative programming languages (examples: Python, Java, C, C++).
  - Students must zip all the files and submit a single file. The zip file must be named with the student's Lastname and ULID.
- The assignment must be submitted only through Moodle.

---

1. (4+3+3 marks) For the following pseudo-codes, what is the time complexity function ($T(n)$) and the order (Θ)? You can ignore the overhead operations and just count the basic operations.

   (a)
   ```
   j = 1 ;
   while ( j <= n/3) {
       i = 1 ;
       while ( i <= j ) {
           cout << j << i ;
           i++;
       }
       j++;
   }
   ```

(b)
```
int F1 (int n)
{ if (n>1)
    {
        return 2*F1(n/2) + (n/2)*(n/2);
    }

}
```

(c)
```
int F1 (int n)
{ if (n>1)
    {
        return F1(n/2) + F1(n/2)+ (n/2)*(n/2);
    }

}
```

2. (4 marks) Is $2n^6 + 3n^5 + 4n^4 + 5n^3 + 6n^2 + 7n + 8 \in o(n^7)$? Prove?

3. (4 marks) Is $n^4 \in o(2n^6 + 3n^5 + 4n^4 + 5n^3 + 6n^2 + 7n + 8)$? Prove?

4. (10 marks) Using the definitions of O and Ω, show that

   (a) $5n^3 + 3100n^5 \in \Theta(n^5)$

   (b) $3n^7 + 4n^6 + 5n^5 + 6n^4 + 7n^3 + 8n^2 + 9n + 10 \in \Theta(n^7)$

5. (12 marks) Solve the following recurrence relations and give a Θ bound for each of them (you do not need to show/prove how the Θ bound is obtained):

   (a) $T(n) = T(n-1) + n$ (assume $T(0) = 0$)

   (b) $T(n) = 2T(n-1) + 3$ (assume $T(0) = 0$)

   (c) $T(n) = 2T(n/2) + n$ (assume $T(1) = 1$)

6. (3+3+2+2 marks) Implement both recursive and iterative algorithms of Fibonacci Sequence shown in class. Determine the largest number that the recursive algorithm can accept as its argument and still compute the answer within 60 seconds. Report how long it takes the iterative algorithm to compute this answer.