Data Structures & Algorithms LAB – Fall 2015

(BS-SE-F14 Morning & Afternoon)

Assignment # 1

Submission Deadline: Wednesday, 28th October 2015 (at the start of class)

Instructions

- **This is an individual assignment. You are NOT allowed to work/submit in form of group. Absolutely NO collaboration is allowed. Any traces of plagiarism/cheating would result in an "F" grade in this course.**
- Only **handwritten** assignments will be accepted.
- Late submissions will NOT be accepted, in any case.
- Clearly mention your **Name**, **Roll Number** and **Section** on the front page of your assignment.

---

1. Show/prove that the following inequalities are correct:

a) $5n^2 - 6n = \Theta(n^2)$

b) $n! = O(n^n)$

c) $2n^2 2^n + n \log n = \Theta(n^2 2^n)$

d) $\sum_{i=0}^{n} i^2 = \Theta(n^3)$

e) $n^3 + 10^6 n^2 = \Theta(n^3)$

2. An algorithm takes **0.5 ms** for input size **100**. How long will it take for input size **500**, assuming that the step count $T(n)$ of the algorithm is:

a) $T(n) = n$

b) $T(n) = n \lg n$

c) $T(n) = n^2$

d) $T(n) = n^3$

e) $T(n) = 2^n$

Note: In both of the above questions, you have to show your complete working to get full credit.

PRACTICE QUESTIONS

Following questions (Question # 3, 4, 5) are for your own practice and you are not required to submit these. But I will assume in quizzes and exams that you have solved these questions.

3. Solve the following questions from your textbook (Sahni's book):

Page #	Exercise #	Remarks
60	1	
61	3 (a), (b)	Also determine the time complexity in Big-Oh notation.
61	4 (d)	Also determine the time complexity in Big-Oh notation.
85	2	Write the algorithm in pseudo-code form.

4. Determine the time complexity (in Big Oh notation) for the following code segments:

```
int temp = 0, i, j;
for (i=0; i<n; i=i+3)
{
      for (j=0; j<n; j=j+2)
      {
            temp++;
      }
}
```

```
int temp = 0, i, j;
for (i=1; i<n; i=i*2)
{
      for (j=1; j<n; j=j*4)
      {
            temp++;
      }
}
```

```
int temp = 0, i, j;
for (i=0; i<n*n; i=i+2)
{
      for (j=1; j<n; j=j*3)
      {
            temp++;
      }
}
```

```
int temp = 0, i, j;
for (i=1; i<=n; i++)
{
      for (j=1; j<=n*n; j=j*2)
      {
            temp++;
      }
}
```

5. Suppose that you have a computer that requires **1 minute** to solve problem instances of size ***n* = 1000**. Now, suppose you buy a new computer that runs **1000 times faster** than the old one. What instance sizes can be run in 1 minute, assuming that the step count $T(n)$ of your algorithm is:

a) $T(n) = n$

b) $T(n) = n^2$

c) $T(n) = n^3$

d) $T(n) = 2^n$

e) $T(n) = 10^n$

☺ GOOD LUCK! ☺