EC330 Applied Algorithms and Data Structures for Engineers
Spring 2020

Homework 1

Out: January, 2020
Due: February 3, 2020 (Extended to Feb 4)

This homework has a written part and a programming part. Both are due at 11:59 pm on February 3. You should submit a scanned/imaged copy of your written part, name it YOUR_BU_ID_HW1.pdf (e.g. U12345678_HW1.pdf) and submit it along with your solution to the programming part on Blackboard. Make sure you show the steps. See Problem 6 for more detailed instructions on what to submit for the programming part.

This is an individual assignment. See course syllabus for policy on collaboration.

1. **Sums [15 pt]**
   Provide a closed-form solution to the following problems. Make sure you show the steps.
   a) $\sum_{i=1}^{330}(\frac{1}{3})^i$
   b) $\sum_{i=0}^{\infty}(\frac{2}{9})^i$
   c) $\sum_{i=1}^{N}(i^3 + 3i^2 - 5i + 7)$

2. **Exponents and Logs [15 pt]**
   Simplify the following expressions. Make sure you show the steps.
   a) $x^1 \cdot x^2 \cdot x^3 \cdots x^{330}$
   b) $log_x x^{330x}$
   c) $log_{330}(330^{330} \cdot 330)$

3. **Combinatorics [10 pt]**
   a) How many 33-digit *hexadecimal* numbers do not contain A, B, C or D?
   b) How many integral solutions of $x_1 + x_2 + x_3 = 33$ satisfy $x_1 \geq 5$, $x_2 \geq 2$ and $x_3 \geq -3$?

4. **Induction [15 pt]**
   Consider Fibonacci numbers $F_0, F_1, F_2, \ldots$ defined by the following rule
   $$F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}.$$
   In this problem, we will confirm that this sequence grows exponentially fast and obtain bounds on this growth.
   a) Use induction to prove that $F_n \geq 2^{0.5n}$ for $n \geq 2$.
   b) Find a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$.

5. **Program Understanding [5 pt]**
What is the value of `sum` after the double-loop exits in the following program? Express your answer as a function of `n`. Show your steps.

```
int sum = 0;
for (int i = n; i > 0; i--) {
    for (int j = n - i; j < n; j++) {
          sum = sum + 1;
    }
}
```

6. **Programming [40 pt]**
Make sure to write your name and BU ID in a comment at the top of the program.
Do not include a `main` in your submitted files.

a) You are given an array of integers. Suppose every integer appears more than once in this array except for one. Write a C++ program that finds the integer that occurs only once. **[20 pt]**
Example #1:
  Input: {1, 3, 3}
  Output: 1
Example #2:
  Input: {2, 3, 4, 7, 4, 4, 2, 3}
  Output: 7

You are given a header file and a sample test program. Your job is to implement the function *findSingle*. Your program must compile and run on the lab computers using the command-line interface as follows.
> g++ -std=c++11 (-o myProgram) testProblem6a.cpp Problem6a.cpp
Submit your solution in a single file named *Problem6a.cpp.*

b) Write a C++ program that keeps summing the digits of the product of two positive integers until the sum becomes a single digit. **[20 pt]**
Example #1:
  Input: 12, 32
  Output: 6
  Explanation: 12 x 32 = 384, 3 + 8 + 4 = 15 → 1+5 = 6
Example #2:
  Input: 34, 57
  Output: 3
  Explanation: 34 x 57 = 1938, 1+9+3+8 = 21 → 2+1 = 3

Again, you are given a header file and a sample test program. Your job is to implement the function *sumDigit*. Similar to a), your program must compile and run on the lab computers using the command-line interface as follows.
> g++ -std=c++11 (-o myProgram) testProblem6b.cpp Problem6b.cpp
Submit your solution in a single file *Problem6b.cpp.* The most efficient solution receives 10 bonus points (out of a total of 100 for this homework).