EC330 Applied Algorithms and Data Structures for Engineers
Spring 2020

Homework 2

Out: February 6, 2020
Due: February 16**,** 2020

This homework has a written part and a programming part. Both are due at 11:59 pm on February 16. You should submit the written part on Blackboard and the programming part on Gradescope.
For the written part, you should submit a single PDF file containing either typeset answers or scanned copies of hand-written answers. Make sure you write your answers clearly.
For the programming part, you should make sure your code compile and run on Gradescope. Make sure you write your name and BU ID in a comment at the top of the program, and use clear comments to explain the key steps in your code.

1. Asymptotic Comparison [30 pt]

In each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both (i.e. $f = \Theta(g)$). **[2 pt each]**

	$f(n)$	$g(n)$
a.	$n - 1$	$n - 330$
b.	$n^{2/3}$	$n^{1/2}$
c.	$330n + \log n$	$n + (\log n)^2$
d.	$n \log n$	$330n(\log 330n)$
e.	$\log 330n$	$\log n$
f.	$330 \log n$	$\log(n^3)$
g.	$n^{1.01}$	$n \log^2 n$
h.	$n^2 / \log n$	$n (\log n)^2$
i.	$(\log n)^{\log n}$	$n / \log n$
j.	$\sqrt{n}$	$(\log n)^3$
k.	$n^{1/2}$	$5^{\log_2 n}$
l.	3^n	$n2^n$
m.	$n!$	2^n
n.	$(\log n)^{10}$	$n^{0.1}$
o.	$\sum_{i=1}^{n} i^k$	n^{k+1}

2. Asymptotic Analysis [10 pt]

Prove that $(\log n)^{\log n} = O(2^{(\log n)^2})$. Show all steps.

3. Programming [60 pt]

a) Consider the infinite sequence 0, 10, 1110, 3110, 132110, 1113122110, 311311222110, … In this sequence, the next number in the sequence is generated based on the previous one. Starting with 0, the next number can be read as "one 0", the one after that can be read as "one 1 one 0", and then "three 1's one 0", and then "one 3 two 1's one 0", and then "one 1 one 3 one 2 two 1's one 0", and so on. Note that for the number 132110, we are generating the next number by scanning this number from left and right and keeping track of the *count of consecutive numbers*, e.g., the two consecutive 1's.
Write a program that outputs the ***k*th digit** (the first digit corresponds to k=1) in this sequence. Note that the k^{th} digit is not the k^{th} element. For instance, the 3rd digit is 0 in this sequence, and the 5th digit is 1 in the sequence.

int kthDigit(int k);

Your job is to implement the *kthDigit* function in a file named "Problem3a.cpp". Submit the single file "Problem3a.cpp" on Gradescope. **[30 pt]**

b) Write a program to find the ***k*th smallest element** in the concatenation of two vectors A and B, given that both vectors are *sorted in decreasing order*. You can assume $1 \le k \le A.\text{size}() + B.\text{size}()$.

The function declaration is given below.

int findKthSmallest(vector<int> A, vector<int> B, int k);

For example, for A = [4, 2, 1] and B = [7, 5, 2, -3], the 2nd smallest element in the concatenation of A and B is 1.

Your job is to implement the *findKthSmallest* function in a file named "Problem3b.cpp". Submit the single file "Problem3b.cpp" on Gradescope. The most efficient solution(s) will receive an *extra credit of 10 point*. **[30 pt]**