CS2383 Fall 2021 Assignment 4 (75 marks)

Due Friday, Nov. 19, by 5pm.

- Assignments in MS Word format should be handed in via D2L. Hand in your source code for Q6 via D2L.

1. **(10 marks)** For each of the following two cases, insert the keys in the sequence given into an initially empty AVL tree. Assume that the keys are in an alphabetic order ($a < b < c < ... < y < z$).

   - a,b,c,d,e,f,g,h,i,j,k,l
   - a,v,l,t,r,e,i,s,f,u,n

2. **(10 marks)** Let D is an AVL tree with n entries. Design an efficient algorithm to implement the following operation on D:

   $findAllInRange(k_1, k_2)$: returns the number of entries in D with key k such that $k_1 \leq k \leq k_2$.

   Analyze the time complexity of your algorithm.

3. **(10 marks)** Draw the 11-item hash table that results from using the hash function $h(i) = (2i + 5) \bmod 11$ to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by double hashing. Use $h'(i) = 7 - (i \bmod 7)$ for the secondary hash function. In both hash functions, the argument i is the value of the key being hashed.

4. **(15 marks)** A 3-way merge sort on an array with N elements works as follows: (1) divide the array into 3 sub-arrays of size $\frac{N}{3}$; (2) recursively sort the 3 sub-arrays; (3). merge the 3 sub-arrays together.

   Write pseudo code of the 3-way merge sort and find its time. You may assume that N is a power of 3.

5. **(10 marks)** Give an efficient recursive algorithm for finding the K^{th} largest element in an array of N elements indexed from 1 to N. Make use of the partition function used in quick-sort. Describe your algorithm in pseudo-code. What is the time-complexity of your algorithm?

6. **(20 marks)** Write a Java program to implement the in-place quick sort algorithm. Your program lets the user to input a sequence of integers, stores them in an array, and outputs this sequence of numbers in an ascending order. Hand in your source code via D2L.