CSIEB0100 Data Structures, Fall 2022
Midterm Exam

ID: ________________ Dept:________________ Name: ______________

1. **(30 points)** Based on the problem type in parentheses (True/False (1 point), fill in the Blank (FIB, 2 points), or briefly answer (ANS, 4 points)), answer the following questions.

1-1 What does "Algorithms + Data Structures = Programs" mean?

____

____ (ANS)

1-2 What are the main evaluation criteria we use to characterize and compare the efficiency of algorithms? ________________________________ (FIB)

1-3 In problem solving, why is the asymptotic improvement on algorithms more important than the increase in processor speed?

____

____ (ANS)

1-4 When $f(n)=\Omega(g(n))$, then $f(n)\geq g(n)$ for all n. ________________ (True/False)

1-5 Give an example of a function $f(n)$ such that $f(n) = \Theta(n \log n)$.

____ (FIB)

1-6 What is the big-O complexity of the expression 1+2+3+…+n?

____ (FIB)

1-7 Rearrange the following complexity classes from the most efficient to the least.

n^3, 2^n, $n \log n$, n^2, $n!$, c, 3^n, $n^2 \log n$, $\log n$

____ (FIB)

1-8 What are the **main** differences between arrays and linked lists?

____

____ (ANS)

1-9 A complete binary tree is also a full binary tree. ______________ (True/False)

1-10 What is the smallest and largest number of nodes of a max heap of height 5? Can you generalize it to a max heap of height *k*?

____

____ (ANS)

1-11 In the KMP algorithm for fast substring search, compute the failure function for the pattern "aabaabaadaabaadc".

____

____ (ANS)

2. **(10 points)** For each of the complexity expression below, determine its overall complexity. For example, given expression $2n + 3n$, the overall complexity should be $O(n)$. Briefly explain why.

(a) $100^n + n^{100} + 2n!$

(b) $5n + 100n \log n + 3n^2$

(c) $n^{1.001} + n \log n + 100\log^2 n$

3. **(10 points)** Given two arrays of integers **A[0..n-1]**, **B[0..m-1]** with no repetition as two sets. Write C++ functions to do the following things. (Note: It doesn't have to be optimal. Correct implementation is good enough.)
   (a) A function to compute and return the array as the **union** of A and B.
   (b) A function to compute and return the array as the **intersection** of A and B.
   (c) Write a main program to test your functions.

4. **(20 points)** Extend the template class `List` discussed in the class with the following functions. Write a main program to test your class properly. (Note: Don't need to write the complete template. Just the extended functions and the main program.)
   (a) `int count(Type);` // Count the number of occurrences of the argument in this list.
   (b) `Boolean exist(Type);` // Return TRUE if the argument exists in this list and FALSE otherwise.
   (c) `int replaceALL(Type, Type);` // Search this list and replace every occurrence of the first argument with the second. Return the number of replacements.
   (d) `List<Type> subList(int, int);` // Return the portion of this list between(inclusive) the first and the second arguments as node indexes (the first node has index 0).
   (e) Write a main program to test your functions

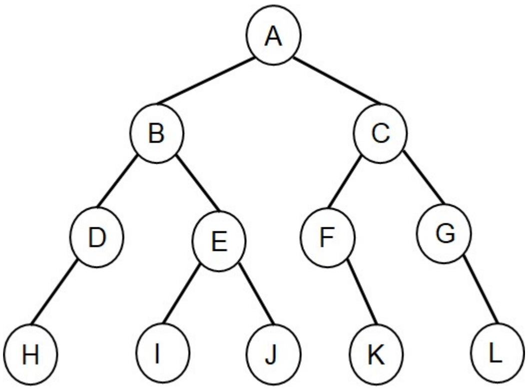
5. **(10 points)** Given the following binary tree, show the results of traversal in various order.

(a) Inorder traversal

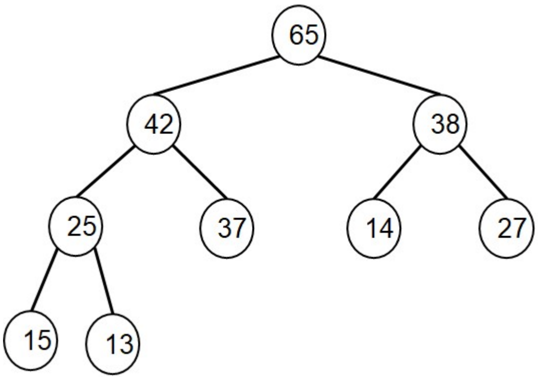
(b) Preorder traversal

(c) Postorder traversal

(d) Level order traversal



6. **(10 points)** On the following max heap, DRAW the resulting max heap after EACH of the following operations: delete, delete, insert 55, insert 35, delete. For delete operations, also show the number that is deleted. (Note: You should draw 5 max heaps and show 3 numbers.)



7. **(10 points)** All linear data structures we studied in class have their pitfalls. Arrays are efficient on indexed access but have size limit and slow on insertion/deletion. Linked lists, stacks and queues are flexible with fast access but only at the end(s). Outline the design of a data structure (to represent a list of items) that is both flexible (with no size limit) and efficient on access/insert/delete operations (at most O(log n).) Answer by drawing your design and explain how the access/insert/delete operations are performed. (Note: No need for C++ code!!)