<div align="center">

CSE, AUA
CS 121 Sample Final Exam
Exam Duration: 2 hours
Exam Total: 100 points

</div>

1. (a) Give the best O-notation for the running time of the following code snippet. Justify your answer.

   int sum $= 0$;
   for(**int** $i = 0; i < n; i = i * 2$)
      for(**int** $j = 0; j < n; j + +$)
         sum= sum +5;

   (b) Order the following 10 functions by asymptotic growth rate.

   $$n!, \ n^{1/3}, \ 2^7, \ 10n^2 + 5\lg n, \ 11n^2, \ 3n\lg n, \ \lg^2 n, \ n^n, \ 6n\lg n + 12\lg n, \ 3^n$$

2. Given two integers n and m, write a recursive function in C++ or Java that returns the greatest common divisor of n and m.

3. Write C++ or Java code that implements the 5 main operations of Queue ADT using two stacks such that the ENQUEUE operation runs in $O(1)$-time.

4. Consider INSERTION-SORT, MERGE-SORT, QUICKSORT and HEAPSORT. For each algorithm, what will be the worst case asymptotic upper bound on the running time if you know additionally that

   (a) the input is already sorted in increasing order?
   (b) the input is sorted in decreasing order?
   (c) the input is an array containing n copies of the same number?

   For each case and each sorting algorithm, state your answer and justify it in one sentence.

5. Consider an array of n integers (not necessarily sorted). Explain how you would find the k-th smallest integer of all n integers in $O(n + k\lg n)$-time. Precisely explain the data structures used and justify the time and space complexities of your method.

6. (a) Describe two advantages of a Binary Search Tree over Hashing.
   (b) Explain two instances where separate chaining is mostly used.

7. Consider an array of small ranged n integers. Give an **efficient** method that counts the number of integer pairs such that one element of the pair divides the other. Explain the running time of your method.

8. Write a function in C++ or Java that reverses a given singly linked list.

9. Consider a binary tree T consisting of n nodes. Write a function in C++ or Java that extracts all leaves of T into a doubly linked list in $O(n)$-time (i.e. doing only a single tree traversal). Explain the time complexity of your function.