CS 2500: Algorithms
Missouri University of Science and Technology
Midterm Exam
Fall 2024

(***Maximum Points: 50***)
(***Time allowed: 65 minutes***)
*(Candidates are allowed additional 10 minutes for **only** reading the paper. They must NOT start writing during this time.)*

This Paper consists of 5 printed pages and 1 blank page.

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

*Answer **five** questions from Part I and **three** questions from Part II.*

Questions are numbered using Arabic numerals $(1, 2, 3, \dots)$*. If the question contains subparts, then you are required to answer **every subpart** of that question. If you have any doubts, please ask the proctor for clarification.*

The intended points for questions or parts of questions are given in brackets [].

You are required to clearly show all steps involved in solving each problem. This includes clearly stating any assumptions made during the problem-solving process, providing step-by-step calculations or logical deductions, and demonstrating how you arrive at your final answer. Each step should be justified with appropriate reasoning, explanations, or theorems where applicable.

Failure to provide clear workings, explanations, or logical reasoning may result in a deduction of points, even if the final answer is correct.

Plan your time wisely. Do not spend too much time on any one problem. Read through all of them first and attack them in the order that allows you to make the most progress.

Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.

Important: *Please ensure that you write (1) full name, (2) student ID, and (3) page number on **EVERY** answer sheet provided. Sheets may get separated during grading.*

ALL THE BEST!

PART I [20 Points]

*Answer **any five** questions.*

1. A sorting technique is called stable if: [4]

   (a) It takes $O(n \log n)$ time

   (b) It maintains the relative order of occurrence of non-distinct elements

   (c) It uses divide and conquer paradigm

   (d) It takes $O(n)$ space

2. Consider the following array:

23	32	45	69	72	73	89	97

   Which of the following options uses the least number of comparisons (among the array elements) to sort the above array in ascending order? [4]

   (a) Selection sort

   (b) Merge sort

   (c) Insertion sort

   (d) Quick sort using the last element as pivot

3. Consider a recursive algorithm that makes two recursive calls for each input of size n and has a space complexity of $O(n)$. If each recursive call consumes $O(1)$ additional space, what is the resulting space complexity? [4]

   (a) $O(n)$

   (b) $O(n \log n)$

   (c) $O(\log n)$

   (d) $O(n^2)$

4. Which of the following is not $O(n^2)$?

   (a) $15 \cdot n^2$

   (b) $n^{1.98}$

   (c) $\frac{n^3}{\sqrt{n}}$

   (d) $20 \cdot n^2$

5. Consider a complete binary tree where the left and the right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is: [4]

(a) $\Omega(\log n)$

(b) $\Omega(n)$

(c) $\Omega(n \log n)$

(d) $\Omega(n^2)$

6. Which of the following correctly determines the solution of the recurrence equation below: [4]

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2T\left(\frac{n}{2}\right) + n & \text{if } n > 1. \end{cases}$$

(a) $\Theta(n)$

(b) $\Theta(n \log n)$

(c) $\Theta(n^2)$

(d) $\Theta(\log n)$

7. An array of 25 distinct elements is to be sorted using quick sort. Assume that the pivot element is chosen uniformly at random. The probability that the pivot element gets placed in the worst possible location in the first round of partitioning is: [4]

(a) $\frac{1}{10}$

(b) $\frac{2}{25}$

(c) $\frac{1}{5}$

(d) $\frac{3}{50}$

PART II [30 Points]

*Answer **any three** questions.*

1. (a) Design a recursive algorithm to compute the Greatest Common Divisor (GCD) of two integers using the Euclidean algorithm. The GCD of two integers a and b is the largest integer n that divides both a and b without leaving a remainder. Write the algorithm in pseudo-code and determine its time complexity in Big-O notation. [5]

   (b) A technology company is assessing the total energy consumption required to run a series of increasingly complex computational tasks. The total energy consumption after n tasks is given by:

$$E(n) = \sum_{i=1}^{n} 2^i$$

   Prove that $E(n) = O(2^n)$. Discuss how the total energy consumption grows as n increases, especially for large n. [5]

2. In a data center, there are 2^n servers labeled from Server 1 to Server 2^n. Each server needs to be updated with the latest security patches. Due to resource constraints, the time it takes to update Server j is inversely proportional to its label number, meaning that Server j takes $\frac{1}{j}$ hours to update.

   (a) Express the total time T required to update all 2^n servers in terms of the harmonic number H_{2^n}. [2]

   (b) Use mathematical induction to prove that:

   $$H_{2^n} \geq 1 + \frac{n}{2}$$

   for all nonnegative integers n. [5]

   (c) Based on your proof, determine the minimum total time required to update all servers when $n = 4$. Explain how the total update time grows as n increases. [3]

3. (a) Assume that the analysis of an algorithm yields $g(n)$ as follows:

   $$g(n) = \sum_{i=1}^{n} 4i(i+3)$$

   Determine the total number of operations and provide the time complexity of the algorithm in Big-O notation. [5]

   (b) Consider the following recurrence relation:

   $$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

   i. Solve the recurrence using the Recurrence Tree Method. Draw the recurrence tree and show your calculations at each level. [3]

   ii. Find the time complexity of $T(n)$ using your solution from the Recurrence Tree Method. Clearly state the final time complexity in Big-O notation. [2]

4. You are a data analyst at a music streaming company. The company has collected a list of the genres of songs played by users over the past month. They want to identify the genre that was played most frequently to tailor their recommendations and promotional efforts. If multiple genres have the same highest play count, any one of them can be considered.

   (a) Design a recursive algorithm that finds the genre that appears most frequently in the list of song genres. [3]

   (b) Analyze the time complexity of your recursive algorithm by deriving its recurrence relation. [3]

   (c) Prove the correctness of your algorithm using mathematical induction. [4]

5. A biotechnology firm is developing a recursive algorithm to simulate the growth of a bacterial colony. The simulation works as follows: If the colony size is zero (i.e., no bacteria present), the simulation time is zero. If the colony size is n, the algorithm splits the colony into two equal groups (assuming n is even for simplicity). It then simulates the growth of each group three times independently due to varying environmental factors. After the simulations, it combines the results, which takes time proportional to n. The time complexity $T(n)$ of the simulation for a colony of size n can be described by a recurrence relation.

   (a) Write the recurrence equation for $T(n)$ based on the simulation process described above. [2]

   (b) It is known that the general closed-form solution to this recurrence is $T(n) = 3n^{\log_2 3} - 2n$, and that $T(n) \in \Theta(n^{\log_2 3})$ This solution is based on the assumption that $T(n)$ is a non-decreasing function. Use mathematical induction to prove that $T(n) = 3n^{\log_2 3} - 2n$ is a non-decreasing function of n. [8]