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

(***Maximum Points: 50***)
(***Time allowed: 90 minutes***)

This Paper consists of 12 printed pages and 1 blank page.

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

Answer ***five*** *questions from Part I and* ***four*** *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 (****except for MCQs.****). 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.

Whenever a question specifically asks for an algorithm, it MUST be written in pseudocode format. Failure to adhere to this will result in zero points for that part of the question.

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.

ALL THE BEST!

PART I [10 Points]

Answer ***any five*** *questions.*

1. (2 points) Which of the following correctly determines the solution of the recurrence equation below:

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

   A. $\Theta(n)$

   B. $\Theta(n \log n)$

   C. $\Theta(n^2)$

   D. $\Theta(\log n)$

2. (2 points) Which of the following algorithm design technique is used in merge sort?

   A. Greedy method

   B. Backtracking

   C. Dynamic programming

   D. Divide-and-Conquer

3. (2 points) Which of the following problems is typically solved using a greedy algorithm?

   A. Longest Common Subsequence

   B. 0/1 Knapsack Problem

   C. Huffman Encoding

   D. All-Pairs Shortest Path

4. (2 points) Among the following sorting algorithms, identify the ones that are not stable:

   (a) Selection Sort

   (b) Quick Sort

   (c) Bubble Sort

   (d) Insertion Sort

   (e) Heap Sort

   Choose the correct option from the choices below:

   A. (a) and (d) only.

   B. (b), (e), and (a) only.

   C. (c) and (d) only

   D. (b) and (e) only.

5. (2 points) Which of the following is a typical step in solving a problem using dynamic programming?
   - A. Sorting the input data
   - B. Building a table to store results of sub-problems
   - C. Always choosing the maximum element first
   - D. Merging solutions of sub-problems

6. (2 points) In Dijkstra's algorithm, what is the correct formula to update the distance to a neighboring node v from the current node u?
   - A. $\text{distance}[v] = \text{distance}[u] + w(u, v)$
   - B. $\text{distance}[v] = \min(\text{distance}[v], \text{distance}[u] + w(u, v))$
   - C. $\text{distance}[v] = \max(\text{distance}[v], \text{distance}[u] + w(u, v))$
   - D. $\text{distance}[v] = \text{distance}[u] + w(v, u)$

7. (2 points) What is the key difference between Prim's and Kruskal's algorithms?
   - A. Prim's algorithm builds the MST incrementally, while Kruskal's processes edges in sorted order
   - B. Kruskal's algorithm works only for connected graphs, while Prim's does not
   - C. Prim's algorithm requires sorting edges, while Kruskal's does not
   - D. Kruskal's algorithm uses a heap, while Prim's uses a union-find structure

---

PART II [40 Points]

*Answer **any four** questions.*

1. In bioinformatics, comparing DNA sequences is a crucial task for understanding genetic similarity between organisms. One common method for comparing sequences is by finding the Longest Common Subsequence (LCS) between two DNA strands.

   You are analyzing two DNA sequences from different species:

$$A = \text{“ACG”}$$

$$B = \text{“TACG”}$$

   (a) (3 points) Derive the recursive formula for solving the LCS problem. Clearly define the base case and recursive relation.

(b) (2 points) Write an algorithm (using dynamic programming) to compute the LCS for two DNA sequences.

(c) (3 points) Apply your algorithm to the given DNA sequences A and B. Determine the length of the LCS and provide one possible LCS. Show the dynamic programming table.

(d) (2 points) Analyze the time and space complexity of your algorithm.

2. You are a data analyst at Spotify. 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.

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

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

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

3. Given: Job A ($t_A = 3$ units), Job B ($t_B = 8$ units), Job C ($t_C = 2$ units), Job D ($t_D = 5$ units). Find an optimal order to schedule the jobs that minimizes the total turnaround time (sum of waiting times plus processing times for all jobs).

   (a) (3 points) Calculate the total turnaround time for the following two specific orders. **If additional space is needed, please use the space for Part (b) below. Ensure you clearly indicate the question and part you are answering to avoid any confusion.**

      1. C, A, D, B (jobs sorted in ascending order of processing time)
      2. B, D, A, C (jobs sorted in descending order of processing time)

(b) (4 points) Identify the order that results in the minimum average turnaround time among the given orders, and explain your reasoning.

(c) (3 points) Explain why scheduling jobs in ascending order of processing time minimizes the average turnaround time, using mathematical reasoning or an example.

4. A communication company needs to efficiently transmit messages over a network using data compression techniques. You are tasked with constructing a Huffman code for the following set of characters and their probabilities of occurrence:

Character	Probability
A	0.35
B	0.1
C	0.2
D	0.2
E	0.15

(a) (6 points) Construct a Huffman code for the given characters. Show all steps involved in building the Huffman tree and assigning binary codes to each character. Place the smaller frequency node as the left child.

(b) (4 points) Using the generated Huffman code, encode the message `ABCAD`. Show your encoding process clearly.

5. A disaster relief team is tasked with distributing supplies to a remote area affected by a hurricane. The team has a truck with a maximum weight capacity of 50 kg, and they have the following items available for transport:

Item	Weight (kg)	Value (importance score)
A	10	60
B	20	100
C	30	120

The goal is to maximize the total importance score of the items transported, while ensuring that the total weight does not exceed the truck's capacity. The team cannot take fractional items, as each item is indivisible.

(a) (3 points) Formulate the problem as a mathematical optimization task. Clearly define what needs to be maximized and the constraints.

(b) (4 points) Write an algorithm to solve this problem and explain how it determines the optimal selection of items.

(c) (3 points) Apply your algorithm to the given data and find the maximum importance score achievable within the weight capacity. Clearly show which items are selected.

6. A disaster has struck a city, and emergency response teams need to determine the shortest path to reach all affected zones from their central base. The city's road network has several roads, but some roads are damaged and have negative travel times due to detours and special conditions (e.g., temporary fast lanes for rescue vehicles).

   The table below represents the road network, where "Base" refers to the central starting point for the emergency response team:

Road (From-To)	Travel Time (minutes)
Base - A	5
Base - B	2
A - B	-3
A - C	6
B - C	3
B - D	7
C - D	-2

(a) (3 points) Formulate the problem as a shortest path problem. Define the input (nodes, edges, and weights) and describe the challenges posed by the presence of negative travel times.

(b) (4 points) Write an algorithm to determine the shortest path from the base to all other zones in the city. Ensure your algorithm handles the possibility of negative travel times appropriately.

(c) (3 points) Apply your algorithm to the given road network. Compute the shortest path from the base to each zone and report the travel times to zones A, B, C, and D.