End Semester Examination

Algorithms Analysis and Design
IIIT Hyderabad, Monsoon 2025

November 27, 2025

There are 10 questions 10 marks each. Maximum Marks: 100. Time: 180 min

---

1. Answer the following regarding Greedy/DP Algorithms: $5 + 5 = 10$ marks

    1. You are an adventurer with a knapsack of capacity W. There are n items, where the i^{th} item has weight w_i and value v_i. You possess a special discount coupon that allows you to reduce the weight of exactly one item in your knapsack to zero. A student suggests sorting items by value-to-weight ratio (v_i/w_i), applying the coupon to the item with the highest ratio, and then filling the rest using the standard greedy approach. Prove or disprove that this strategy succeeds/fails in always arriving at the optimal output. In case you think/proved that greedy fails, provide a correct dynamic programming solution.

    2. The Highway Patrol wants to cover a highway of length L. You are given n potential patrol segments. Segment i covers the interval $[s_i, f_i)$ and has a strategic value v_i. You must select a subset of segments such that no two segments overlap. Prove or disprove whether the "`Earliest Finish Time`" greedy strategy (commonly used for unweighted interval scheduling) works/fails to maximize the total value v_i in this variant. In case you think/proved that greedy fails, provide a correct dynamic programming solution.

2. Answer the following regarding Divide-and-Conquer Algorithms: $4 + 3 + 3 = 10$ marks

    1. You are given two sorted arrays A and B, each of size n. Design an $O(\log n)$-time algorithm to find the median of the combined set $A \cup B$.

    2. You are given a set of n buildings, each represented by a coordinate pair (x_i, h_i), where x_i is the location and h_i is the height. A building i is said to dominate building j if: $x_i < x_j$ and $h_i > h_j$ (i.e., building i is to the left of and taller than building j). Design a $O(n \log n)$-time algorithm to count the total number of dominance pairs (i, j) in the set.

    3. You are given a binary text string T of length n and a binary pattern P of length m, assume $m \leq n$. Both strings consist only of $\{0, 1\}$. You want to find all indices i in T such that the substring $T[i \ldots i+m-1]$ matches P with at most k mismatches (Hamming distance $\leq k$). Formulate this problem using polynomials and describe how to solve it in $O(n \log n)$ time using Fast Fourier Transform (FFT).

3. Answer the following regarding Number-Theoretic Algorithms: $1+2+2+3+2=10$ marks

   1. Given a product of two primes, $N = pq$, and further given $\Phi(N)$ (the Euler totient function), show that N can be ***efficiently*** prime factorized.
   2. More generally, do you think that for any N (not just product of two primes), giving $\Phi(N)$ along with N enables you to efficiently factorize N?
   3. Let N_1, N_2 and N_3 be distinct RSA moduli, such that $gcd(3, \Phi(N_1)) = gcd(3, \Phi(N_2)) = gcd(3, \Phi(N_3)) = 1$. Let $e = 3$. Show that, given three vanilla RSA ciphertexts of a number/message $m < min(N_1, N_2, N_3)$ under public keys (N_1, e), (N_2, e) and (N_3, e) respectively (that is, $c_j = m^3 \bmod N_j$), one can efficiently find the message m.
   4. More generally, suppose $N_1, N_2, \ldots, N_k$ are distinct RSA moduli, such that $gcd(e, \Phi(N_1)) = gcd(e, \Phi(N_2)) = gcd(e, \Phi(N_3)) = 1$ for some $e \leq k$. Do you think given e vanilla RSA ciphertexts of a number/message $m < min(N_1, N_2, \ldots, N_k)$ under public keys (N_1, e), $(N_2, e), \ldots, (N_k, e)$ respectively, you can efficiently find the underlying message m?
   5. Describe an efficient algorithm that given two relatively prime numbers e and M, finds a d such that $ed \equiv 1 (\text{mod } M)$. Illustrate your algorithm for $M = 264$, $e = 5$.

4. Answer the following regarding Distributed Algorithms: $3+2+2+3=10$ marks

   1. Prove that among three synchronous fully connected parties (perfect) Byzantine Agreement (BA) is impossible even with a single Byzantine fault.
   2. Use the above to prove that $n > 3t$ is necessary for perfect BA among n synchronous fully connected parties, where up to t among them are Byzantine corrupt.
   3. Design a BA protocol that works for $n = 4$ and $t = 1$, over a synchronous networks with six edges (complete graph on 4 nodes).
   4. Prove that it is impossible to achieve perfect BA among 4 parties with one Byzantine fault if the synchronous network has ≤ 4 undirected edges (out of the possible 6 edges). Also, show that ≥ 5 edges suffice!

5. Answer the following regarding Quantum Algorithms: $1+2+3+4=10$ marks

   1. State and prove the No-Cloning Theorem.
   2. Describe how to establish a secret-key between two users using quantum mechanics but without entanglement (like BB84 protocol). Argue why is the No-Cloning theorem required for security of your protocol.
   3. Present the protocol for quantum teleportation of a qubit and argue that is does not violate the No-Cloning Theorem.
   4. Illustrate how you may efficiently break the RSA cryptosystem using a quantum computer using Shor's algorithm.

6. Answer the following regarding Randomization and negligibility: $1+2+3+4=10$ marks

   1. Formally define when is a function *negligible*.
   2. Which of the following functions are negligible (prove your answers).

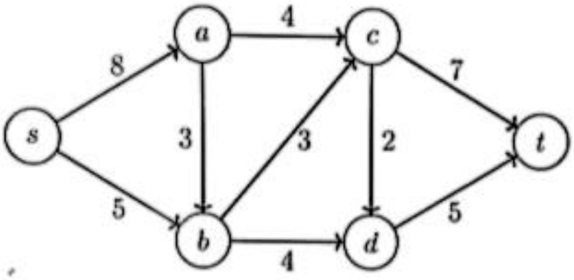
(a) $f(x) = 2^{-40}$ (that is, one part in a trillion).
(b) $f(x) = \frac{1}{x}$.
(c) $f(x) = \frac{1}{(\log x)!}$.
(d) $f(x) = \frac{1}{(\log \log x)!}$.

3. Illustrate the Miller-Rabin (randomized) primality testing algorithm to test if 257 is a prime or not? Are you sure of the answer?

4. For the input 21 compute the exact number of witnesses (of compositeness) and non-witnesses in the range $[2, 20]$ for the Miller-Rabin primality testing algorithm.

7. Answer the following regarding Network Flows and Spanning Trees: $3 + 2 + 2 + 3 = 10$ marks

1. Compute the maximum flow from s to t in the following network. Also point out some minimum-cut that corresponds to the maximum flow.



13

2. What is a minimum spanning tree for the above network/graph (work with the underlying undirected graph, that is ignore the directions in the edges).

3. Let $G = (V, E)$ be a flow network with source s, sink t, and integer capacities. Suppose that we are given a maximum flow in G. Suppose that we increase the capacity of a single edge $(u, v) \in E$ by 1. Give an $O(V + E)$-time algorithm to update the maximum flow.

4. A telecommunications company has built a network using a Minimum Spanning Tree (MST) T of a graph $G = (V, E)$ to minimize wiring costs. They require a backup plan. Design an algorithm to find the "Second Best MST." Formally, find a spanning tree T' such that $T' \neq T$, and the sum of weights $w(T')$ is minimized among all spanning trees other than T. Analyze the complexity. Illustrate your solution on the above drawn network/graph (used in part (2) of this question where the MST T is found).

8. Answer the following regarding Computability Theory, NP-Hardness and design and analysis of Approximation Algorithms: $2 + 3 + 2 + 3 = 10$ marks

1. Let $EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$, that is check if two given Turing machines have the same language. Prove that EQ_{TM} is neither recognizable nor co-recognizable (that is $EQ_{TM} \notin RE$ and $\overline{EQ_{TM}} \notin RE$).

2. Assuming 3-SAT is NP-complete prove that computing the maximum sized clique in a graph is NP-Hard. Also prove the NP-Hardness of minimum vertex-cover problem.

3. Both minimum vertex-cover and maximum clique problems are NP-hard (as proved by you above). Does this imply that the 2-approximate algorithm for minimum vertex can be adapted to achieve constant approximation ratio for the clique problem? Justy your answer.
4. Design and analyze the approximation ratio of a greedy algorithm to solve the MINIMUM SET COVER problem.

9. Answer the following regarding Dynamic Programming: $4 + 2 + 4 = 10$ marks

1. Design an algorithm to find the length of the Longest Common Subsequence (LCS) of *three* strings X, Y, and Z of lengths n, m, k respectively. Analyze the time complexity.
2. A student claims that LCS(X, Y, Z) can be computed by finding W = LCS(X, Y) first, and then computing LCS(W,Z). Prove or disprove this claim with a counter-example.
3. Given a tree $T = (V, E)$ where every vertex v has a weight w_v. An Independent Set is a subset of vertices where no two vertices share an edge. Design a linear time $O(V)$ algorithm to find the Independent Set with the maximum total weight.

10. Answer the following regarding your course project: $4 + 2 + 2 + 2 = 10$ marks

1. Briefly describe your course project.
2. What is your role and contributions to the project?
3. What parts of the project are enjoyable to you, and parts where you felt otherwise?
4. What could be the potential future directions to extend your project. Justify.

Jareth vs Chan's

ALL THE BEST!