MidSemester Examination - September 2026
Design and Analysis of Algorithms

Time: 2 Hours
Max Marks:40

Instructions

- *When you give an algorithm, give a full analysis including proof of correctness.*
- *A greedy algorithm will not be accepted without a proof of correctness.*
- *When a dynamic programming algorithm is presented, follow the given procedure.*
  - *Define sub-problem(s) and explain how they help in solving the given problem.*
  - *Design a recurrence relation for the sub-problem and prove its correctness.*
  - *Explain the order in which the sub-problems are computed.*

Good Luck!!

1. Given an array of n elements, design a divide and conquer algorithm with time complexity $O(n \log n)$ to determine whether there exists an element that repeats more than $\frac{n}{3}$ times in the array.**(7 marks)**

   The elements are not comparable; only equality checks between elements are allowed.

2. Given an array A with n distinct numbers, a modified MergeSort algorithm works as follows : A_1 is the subarray $A[1 \ldots \lceil \frac{n}{3} \rceil]$, A_2 is the subarray $A[\lceil \frac{n}{3} \rceil + 1 \ldots \lceil \frac{2n}{5} \rceil]$ and A_3 is the subarray $A[\lceil \frac{2n}{5} \rceil + 1 \ldots n]$. Recursively sort A_1, A_2, A_3. Merge the sorted subarrays and return the final array. How does the asymptotic running time of this modified MergeSort compare with that of the MergeSort algorithm discussed in class? Justify.**(3 marks)**

3. You have to schedule n computing jobs. Every job needs to be ”preprocessed” on a supercomputer first and then run on a desktop computer. For every job i, the time needed on supercomputer is f_i and the desktop is s_i. You have limited access to supercomputer so only one job can run on the supercomputer at a time. However any number of jobs can run in parallel on desktop computers. Given the values of f_i and s_i for $1 \leq i \leq n$, find a schedule of the jobs so that the end time of the last job to finish is minimized. **(10 marks)**

4. Recall the greedy algorithm discussed in the class to solve the following problem.
   Given a set of m intervals $\mathcal{I}$, set of n points P on the real line such that every point is contained in at least one interval in $\mathcal{I}$, find a subset $\mathcal{I}'$ of $\mathcal{I}$, such that for every point $p \in P$, there exists at least one interval $I \in \mathcal{I}'$ such that $p \in I$ and $|\mathcal{I}'|$ is minimised.
   Now, say whether the same algorithm works for the following modified problems. If yes, give a brief justification. Otherwise, give a counterexample.**(10 marks)**

   (a) The input contains a weight function $w : \mathcal{I} \to \mathbb{N}$ and the output requires a subset $\mathcal{I}'$ of $\mathcal{I}$, such that for every point $p \in P$, there exists at least one interval $I \in \mathcal{I}'$ such that $p \in I$ and $\sum_{I \in \mathcal{I}'} w(I)$ is minimised.

   (b) The input remains the same and the output requires a subset $\mathcal{I}'$ of $\mathcal{I}$, such that for every point $p \in P$, there exists *exactly* one interval $I \in \mathcal{I}'$ such that $p \in I$.

   (c) The input remains the same and the output requires a subset $\mathcal{I}'$ of $\mathcal{I}$, such that for every point $p \in P$, there exists at least one interval $I \in \mathcal{I}'$ such that $p \in I$ and no point is contained in more than two intervals in $\mathcal{I}'$.

5. A metapalindrome is a decomposition of a string into a sequence of palindromes, such that the sequence of palindrome lengths is itself a palindrome. Describe and analyze an efficient algorithm to find the length of the shortest metapalindrome for a given string. **(10 marks)**