INDIAN STATISTICAL INSTITUTE

MIDTERM EXAMINATION
M.TECH(CS) I YEAR

DESIGN AND ANALYSIS OF ALGORITHMS

Date: 28.02.2025 Maximum marks: 60 Duration: 2.5 hours.

Unless explicitly asked, you can use any standard algorithm discussed in class without description. The paper contains questions worth 70 marks. Answer as much as you can, the maximum you can score is 60.

1. (a) Describe a strategy to modify the Quicksort algorithm so that it runs in $O(n \log n)$ time in the worst case.

   (b) Given a sorted array A of size n, describe an algorithm to determine if there is an i, such that $A[i] = i$. What is the running time of your algorithm?

   (c) An element a of an array A of size n is called a majority element if a occurs in A at least $\lceil n/2 \rceil$ times. Let $\mathsf{Count}(A, a)$ be a linear time procedure which on input an array A and an element a outputs the number of times a occurs in A. Consider the following randomized algorithm which finds a majority element in an array A of size n:

```
Algorithm MAJORITY(A)
1.  for i = 1 to 50
2.      j ←$ {1, 2, 3, ..., n}
3.      m ← Count(A, A[j])
4.      if m ≥ ⌈n/2⌉
5.          return "A[j] is a majority element"
6.      end if
7.  end for
8.  return "There is no majority element in A"
```

   What is the asymptotic running time and the probability of error of the above algorithm.

   (d) Using a suitable adversarial argument show that for finding the minimum of a distinct set of n integers the number of comparisons necessary is $n - 1$.

   (e) Given three arrays A, B, C, each containing n integers in the range $[0, 10n]$, give an efficient algorithm to find whether some $a \in A$, some $b \in B$, and some $c \in C$ have zero sum, i.e., $a + b + c = 0$. To get any credit, the asymptotic running time of your algorithm should be better than $O(n^3)$.

$[5 \times 6 = 30]$

2. Recall that the Fibonacci numbers $F_0, F_1, F_2 \ldots$, are defined by the rule

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2} \text{ for } n \geq 2$$

(a) Prove that

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

(b) Using (a) or otherwise describe an efficient algorithm to compute $F_n \bmod p$ for any given n and p. To get full credit your algorithm should run in time $O(\log n \log^2 p)$.

[3 + 7 = 10]

3. Suppose we have t sorted arrays, each with n elements. Describe an algorithm to merge these t arrays to a single sorted array. What is the complexity of your algorithm? You can assume that you have a procedure Merge(L_1, L_2) which when given as input two sorted arrays L_1 and L_2 of size ℓ_1 and ℓ_2 respectively, outputs a sorted array of length $\ell_1 + \ell_2$ in time $O(\ell_1 + \ell_2)$. For full credit, your algorithm should have a asymptotic running time better than $O(nt^2)$.

[7 + 3 = 10]

4. From a sorted array $X = [x_1, x_2, \ldots, x_n]$, an array $Y = [y_1, y_2, \ldots, y_n]$ is created where

$$y_i = x_i^2 - 4x_i.$$

Describe a linear time algorithm to sort Y. Justify your answer.

[10]

5. Given three polynomials $P(x), Q(x), R(x) \in \mathbb{Z}_p[x]$, where P, Q are of degree n and R is of degree $2n$. Assume p is a prime and $p > 2n$. You need to write a procedure verify(P, Q, R) which returns 1 when $P(x)Q(x) = R(x)$ and returns zero otherwise.

(a) Write a deterministic algorithm verify(P, Q, R) which runs in $O(n^{\lg 3})$ time in the worst case.

(b) Write a randomized procedure verify(P, Q, R) with worst case running time $O(n)$. What is the maximum probability of error of the procedure.

[5 + 5 = 10]