**Winter 2024**
**March 15, 2024**

# Algorithm Design and Analysis: Midsem Re-Exam

*Max: 40 Marks*
*75 minutes*

---

**Roll No:** _____ **Name:** _____

*Please write solutions independent of each other. This is a closed book test. You can not use books or lecture notes. Please note that your solution must fit in the space provided. Extra sheet will be provided* only *for roughwork . So try to be precise and brief. Meaningless blabber fetches negative credit.*

| Question | 1 | 2 | 3 | Total |
|---|---|---|---|---|
| Marks | | | | |

Total Marks:

---

1. (10 **Marks**) Given two arrays $A$ and $B$ having $n$ distinct numbers each. Define

$$\mathsf{Pred}(a) = \max_{b \in B}\{b < a\}, a \in A$$

Informally, for every $a \in A$, the $\mathsf{Pred}(a)$ is the maximum number $b \in B$ such that $b < a$. Design an algorithm that computes $\mathsf{Pred}(a)$ for all $a \in A$.

You will be awarded zero marks if the running time of your algorithm is $O(n^2)$ or worse.

(i) Description of your algorithm:

(ii) Explanation of running time:

2. **(20 Marks)** Suppose you are managing the construction of billboards on a highway. The sites of billboards are given by distinct natural numbers $x_1, x_2, \ldots, x_n$ that are in the interval $[0, M]$, i.e. for every $i = 1, \ldots, n$, $0 \leq x_i \leq M$. More specifically, $x_i$ denotes the position of the $i$-th site in this highway. If you place a billboard at location $x_i$, then you get a revenue of $r_i > 0$.

   Regulations of the country says that two billboards cannot be placed less than or equal to 5 miles of each other. You would like to place the billboards that at a subset of sites so that you can maximize the revenue.

   Design a dynamic programming based algorithm that computes the maximum total revenue that can be obtained from any valid subset of sites. The running time of your algorithm must be polynomial in $n$.

   (i) Definition of your subproblem:

   (ii) Recurrence of the subproblem:

   (iii) The subproblem that solves the final problem:

(iv) Algorithm Description:

(v) Explanation of the running time:

3. (**10 Marks**) The *frequency* of a number $x$ in an array $A$ is the number of times $x$ appears in $A$. Given an array $A$ of $n$ numbers, design an algorithm that finds a number in the array $A$ with maximum frequency. Your algorithm must run faster than $O(n^2)$-time. Give a clear explanation of the running time of your algorithm.