

INDIAN STATISTICAL INSTITUTE

MTech(CS) I year 2025-2026

Subject: Computing Laboratory

Lab Test 2 (March 31, 2026)

Total: 60 marks Duration: 4 hours

GENERAL INSTRUCTIONS

1. All programs should take the required input via the standard input (terminal/keyboard), and print the desired output to the terminal.
2. Please make sure that your programs adhere strictly to the specified input and output format. Do not print extra strings asking the user for input, debugging messages, etc. These will cause the automatic checking system to fail.

Problem 1 (5 + 20 = 25 marks)

Let \mathcal{R} denote the rectangular grid bounded by the origin $O(0,0)$ at the lower left corner, and $P(m,n)$ at the upper right corner ($m > 0, n > 0$). Two points A and B are initially at (x_a, y_a) and (x_b, y_b) respectively, within \mathcal{R} . Starting at $t = 0$, each of A and B moves l steps and stops. Each step takes unit time, and involves a move of one unit in either the X or the Y direction, to an adjacent position in \mathcal{R} ; moves to grid points outside \mathcal{R} are not permitted. At every step, all permissible moves are equally likely; A 's and B 's moves are independent.

Q1. Write a function with the following prototype:

```
int enumerate_neighbours(int m, int n, int x, int y, int *neighbours)
```

The function should take the grid size ($m \times n$) and a grid point (x, y) , store a list of the grid points adjacent to (x, y) in `neighbours`, and return the number of such adjacent points.

Input format: Four non-negative integers in the order m, n, x, y .

Output format: A list of the x- and y-coordinates of all grid points adjacent to the given point. The points on the list should be sorted in increasing order of y-coordinates as the primary sort key, and in increasing order of x-coordinates as the secondary sort key.

Sample input 1: 10 10 0 0

Sample output 1: 1 0 0 1

Sample input 2: 10 10 4 4

Sample output 2: 4 3 3 4 5 4 4 5 [5]

Q2. Write a recursive program to compute the probability that A and B meet at least once, i.e., they are both at the same grid position simultaneously at some time $t \in \{0, 1, 2, \dots, l\}$. If A and B are in adjacent grid points, and in the next step, they interchange positions, by definition, this is **not** counted as a meeting.

Input format: Seven non-negative integers in the order $m, n, x_a, y_a, x_b, y_b, l$.

Output format: A single floating point number, correct to 10 decimal places.

NOTE: To avoid small mismatches due to floating point arithmetic, please store all floating point numbers in `double` type variables. [20]

Problem 2 (8 + 15 + 12 = 35 marks)

You are given a non-empty binary tree containing integer values in the “standard” tabular form used in class (so you can directly use the provided `read_tree()` function to read it into memory).

Q3. Add a `depth` field to each node. Write a function that traverses the tree in preorder, fills in the `depth` field of each node, and returns the height of the tree.

Output format: Print the tree in the usual tabular form, along with an additional (4th) column containing the depth. Add one line to the output containing just the height. Thus, your output should have $n + 2$ lines. [8]

Q4. Write a program to do a breadth-first traversal of the tree, and to check whether the following properties hold wherever applicable.

- The integer stored in any internal node \geq any integer stored in any of its children.
- If the height of the tree is h , then leaf nodes can occur only at depth h and $h - 1$.
- At most one non-leaf node can have one child; such a node must occur at depth $h - 1$.

Output format: If all nodes satisfy all applicable conditions, your program should print YES. Otherwise, it should print the index and value of the first node (in breadth-first order) that violates any of the above conditions. [15]

Q5. Write a program to determine whether the tree is a binary search tree.

Output format: If the binary tree is a binary search tree, your program should output YES. Otherwise, your program should print the index and value of the first node in preorder that violates any of the binary search tree properties. [12]