

ASSIGNMENT-1

Jukanti lohith

2303A51528

Batch - 19

Assignment 1: Maximum Non-Overlapping Meetings (Greedy)

Problem Statement

You are given N meetings. Each meeting has a start time S_i and an end time E_i . You want to attend the maximum number of meetings. You can attend meeting j after meeting i only if the start time of meeting j is strictly greater than the end time of meeting i ($S_j > E_i$). For each test case, output the maximum number of meetings that can be attended.

Input Format

The first line contains an integer T, the number of test cases. For each test case:

- The first line contains an integer N.
- The next N lines each contain two integers S_i and E_i .

Output Format: For each test case, print a single integer: the maximum number of meetings that can be attended. Constraints

- $1 \leq T \leq 20$
- $1 \leq N \leq 200000$ (sum of N over all test cases ≤ 200000)
- $0 \leq S_i < E_i \leq 10^9$

Sample Input

```
1
3
1 3
2 4
3 5
```

Expected Output 2

ASSIGNMENT-1

The screenshot shows a Java development environment with the following details:

- IDE Interface:** The top bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". The title bar says "Q COMPUTITIVE PROGRAMMING".
- Left Sidebar:** Includes sections for "RUN AND DEBUG", "VARIABLES", "WATCH", "CALL STACK" (showing threads like "Attach Listener", "Finalizer", "Reference Handler", "Signal Dispatcher", "Notification Thread", "Common-Cleaner" all in "RUNNING" state), and "BREAKPOINTS" (with checkboxes for "Uncaught Exceptions" and "Caught Exceptions").
- Code Editor:** The main area displays the following Java code:

```
A_1.java > Main > main(String[] args)
1 import java.util.*;
2 class Main {
3     static class Meeting {
4         int start, end;
5         Meeting(int s, int e) {
6             start = s;
7             end = e;
8         }
9     }
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        int T = sc.nextInt();
13        while (T-- > 0) {
14            int N = sc.nextInt();
15            Meeting[] meetings = new Meeting[N];
16            for (int i = 0; i < N; i++) {
17                meetings[i] = new Meeting(sc.nextInt(), sc.nextInt());
18            }
19            Arrays.sort(meetings, (a, b) -> Integer.compare(a.end, b.end));
20            int count = 0;
21            int lastEnd = -1;
22            for (Meeting m : meetings) {
23                if (m.start >= lastEnd) {
24                    count++;
25                    lastEnd = m.end;
26                }
27            }
28        }
29        System.out.println(count);
30    }
31    sc.close();
32 }
33
```

- Bottom Status Bar:** Shows "Ln 18, Col 14", "Spaces: 4", "UTF-8", "CR/LF", "Java", "ENG IN", "09:36 AM", and the date "16-01-2026".
- Output Window:** Labeled "PROBLEMS 2", "OUTPUT", "DEBUG CONSOLE" (which is selected), "TERMINAL", and "PORTS". It displays the following text:
 - "Listening on 59384"
 - "User program running"
 - Sequence of numbers: 1, 3, 1 3, 2 4, 3 5.
 - "User program finished"
 - Final number: 2

ASSIGNMENT-1