50 marks **CS 301 Quiz 1** **8:30-9:30 am, 24/8/7**

Problem 1: In this problem, you will devise an algorithm which given n straight line segments in the plane, will determine the number of pairwise crossings, i.e. out of the $\binom{n}{2}$ pairs of lines, how many intersect. You are only to consider a special case of the problem in which one endpoint of every line lies on the x axis, and the other on the line $y = 1$. You may also assume that the endpoints of all lines are distinct. An instance will be specified as a list of pairs where each pair (a, b) denotes a line connecting the points $(a, 0)$ and $(b, 1)$ An example instance is ((1,3), (2,2), (3,1), (4,5), (5,6), (6,7), (7,8), (8,4)) which represents the lines as shown. For this instance the answer is 7, as you can check.

Here is the algorithm skeleton for you to develop.

1. Let x_m denote the median x-coordinate of the upper endpoint of each segment (median for n elements: $\lfloor n/2 \rfloor$ th largest)
2. Colour a line green if its upper end point has x coordinate at most x_m. Otherwise colour the line red.
3. Recursively find G = number of intersections of green lines with each other, and R = number of intersections of red lines with each other.
4. Whatever more is needed. For this it might help to consider the lines in increasing order of the x coordinate of the bottom endpoint.

(a)[5 marks] What are x_m, G, R for the example shown? Also show which lines are green and which are red.

(b)[5 marks] Will the number of intersections be bigger than $G + R$? Why?

(c)[15 marks] Give an algorithm based on the above skeleton. Estimate the time taken by your algorithm.

The base case is with $n = 1$ line. No intersection.

Need to find the intersections between the green and red lines. The following observation is crucial. Scan the lines in increasing order of the x coordinate of the bottom end point. Keep track of the number of red endpoints passed. If the ith endpoint is green and has j red endpoints before it, then the ith green line has exactly $i - j$ intersections with red lines. Thus in a single pass, O(n) time, it is possible to determine the number of intersections for each green line with all red lines. Adding up gives us the result. 8 marks for above.

2 marks to observe that sorting is not needed if the input is sorted.

The time will satisfy T(n) = 2T(n/2) + O(n). Thus T(n)+O(nlogn). 5 marks. You get these 5 marks if you give any correct algorithm above and give the right recurrence. Unless

you make the observation given above, your solution will take $O(n^2)$ time – which is fine in this part. But if you dont make the observation given above and simply write T(n) = 2T(n/2) + O(n) then you dont get these marks.

Problem 2:[25 marks] Suppose there are n projects on which you can work. There are a total of H hours you can spend on these projects. The returns you get on each project are given to you as an $n \times H$ matrix R, where $R[i, j]$ denotes the return (in rupees) you get if you spend j hours on project i. You may choose to spend no time on some of the projects – in which case your returns will be 0 for those projects, of course. The returns need not be linear, i.e. doubling the time spent on a project may not double the return.

Give an algorithm that takes as input integers n, H and the matrix R and determines how many hours to allocate to each project so that the total return (the sum of the returns on each project) is maximized.

Let G[i,H] denote the maximum return for projects i through n using H hours.

$$G[i, H] = \max_{r \in [0,H]} \{R[i, r] + G[i + 1, H - r]\}$$

G[n,j]=R[n,j] for all j. Thus every entry takes time O(H) to fill. Total of nH entries, so time $O(nH^2)$.

Recurrence: 12 marks, 3 for the base case (i=n). Order of filling entries: decreasing i. 5 marks. time analysis: 5 marks.