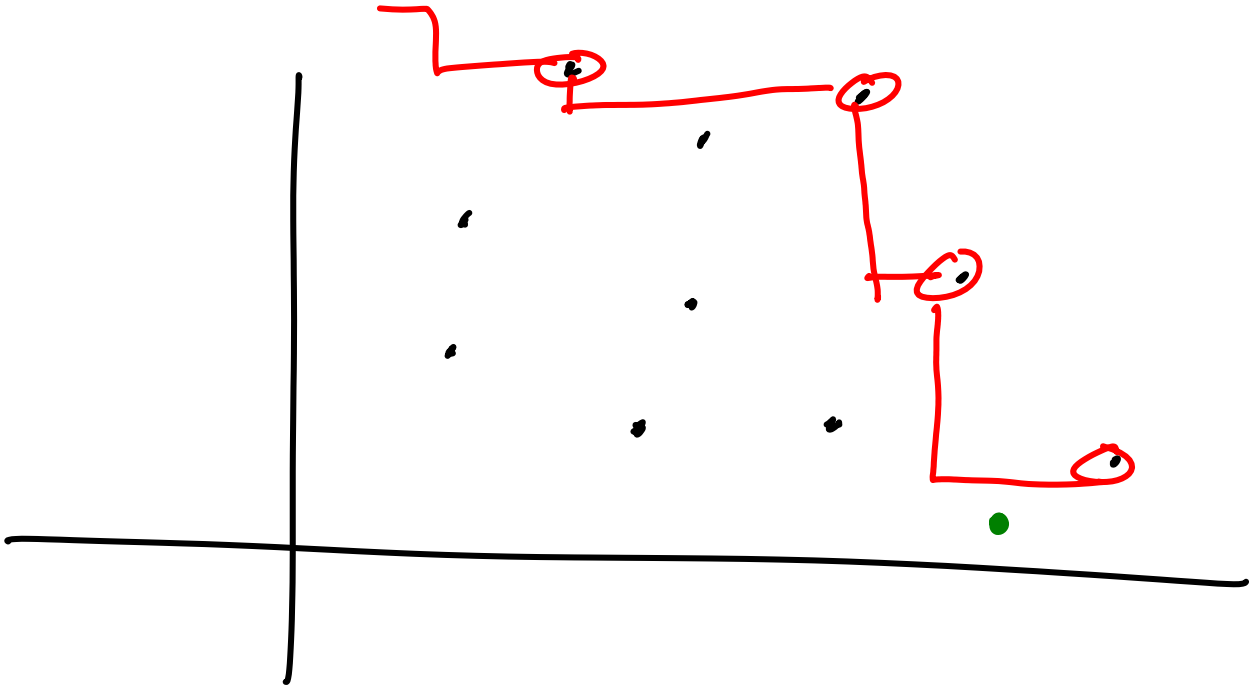
Lecture 2 COL 702

Maximal points

Given a set S of n tuples, find out the maximum subset $V \subset S$ s.t. no tuple in V is dominated by another tuple. A tuple : $(x_1, x_2, x_3 \cdots x_d)$



$O(n^2)$ brute force algorithm is easy : Just do all pairwise comparisons to eliminate the points that are not maximal

Can we do better?

Algorithm I

Step 1: Sort the points on the basis of their x coordinates $O(n \log n)$

Step 2: Scan the points in reverse order of their x coordinates

Keep track of the highest y coordinates among the points scanned, say y_{max} $O(1)$ for each iteration

Step 3 If the present point (x_i, y_i) is such that

$y_i < y_{max}$ then (i-th is not maximal)

else (x_i, y_i) is maximal and

$y_{max} \leftarrow y_i$

until all points are scanned

Proof that the algorithm correctly identifies all the maximal points

By induction on the loop
(loop invariants)

All the points that have been
scanned have been correctly classified

Base case : max. 2 coordinate

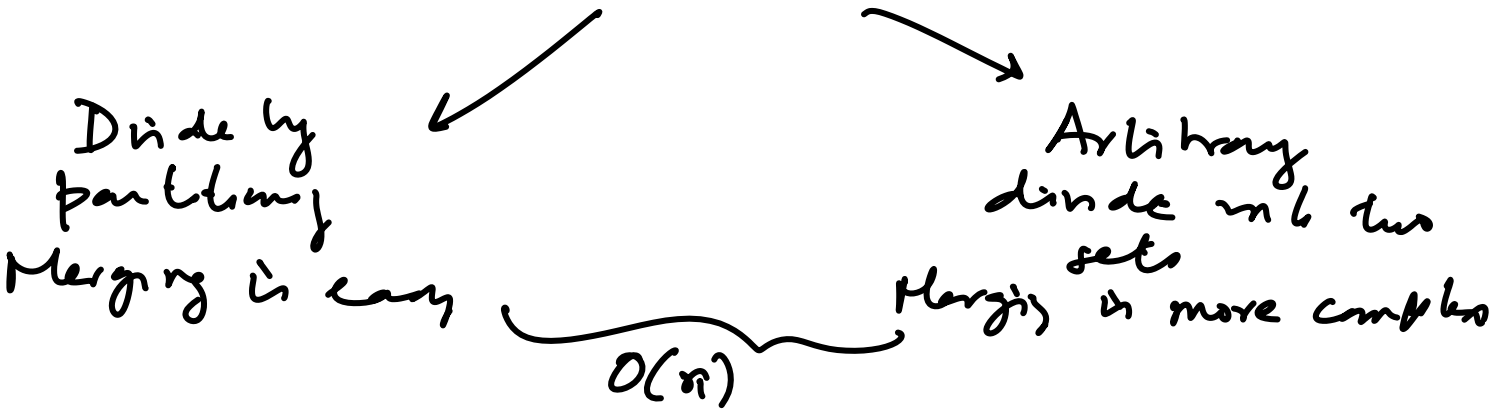
Time and Space Complexity Analysis

$O(n \log n)$ + $O(n)$ → $O(n \log n)$

sorting main loop

Sweep or Line sweep

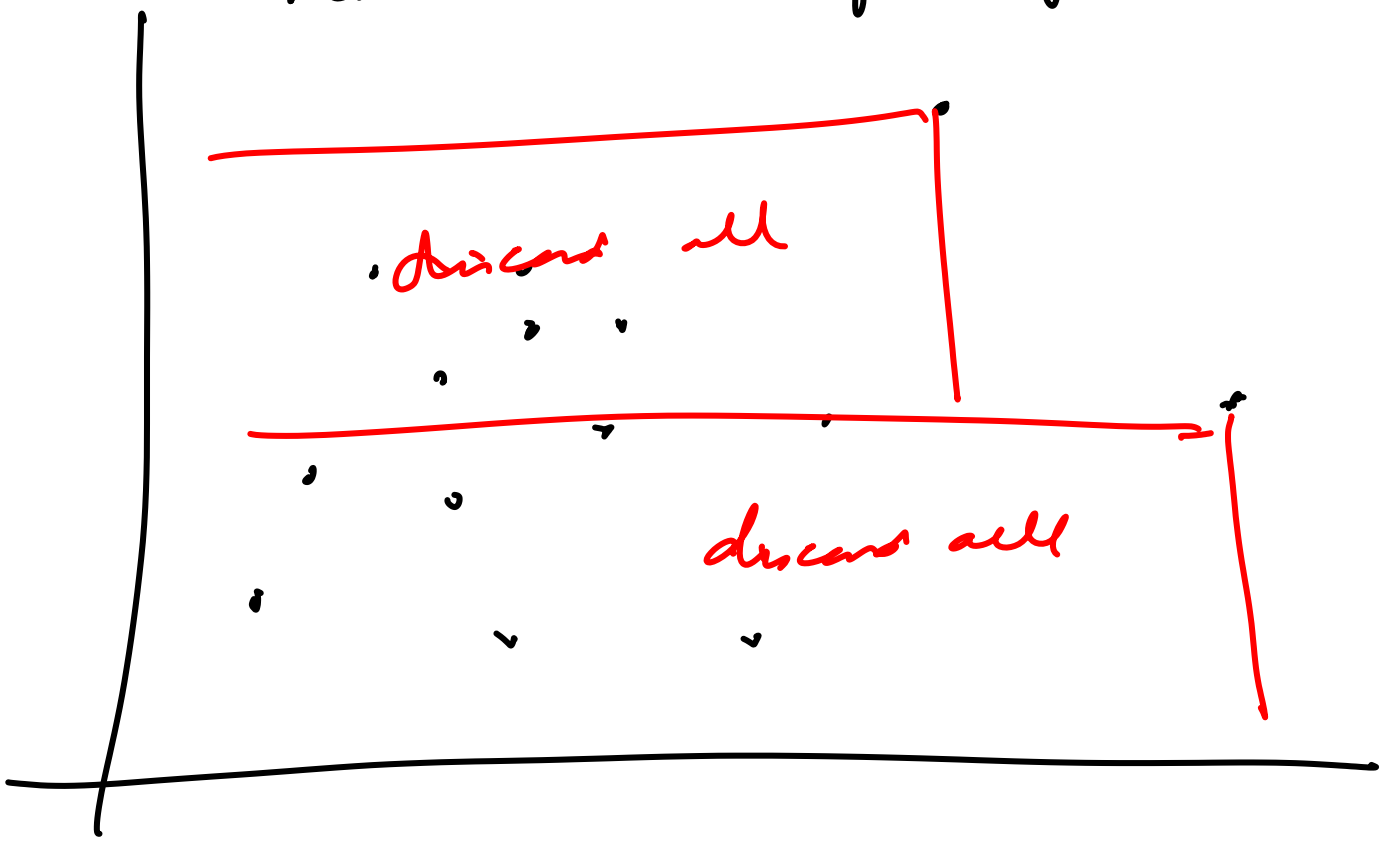
Try Divide-and-conquer



$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$\Rightarrow T(n) = O(n \log n)$$

Consider the following input



By filtering out all the points dominated by the latest maximal point

$$T(n) = h\, n$$

h ↗ # maximal points

Output-sensitive algorithm

Better output-sensitive algorithm

Lower running-time

$$O(n \log h)$$