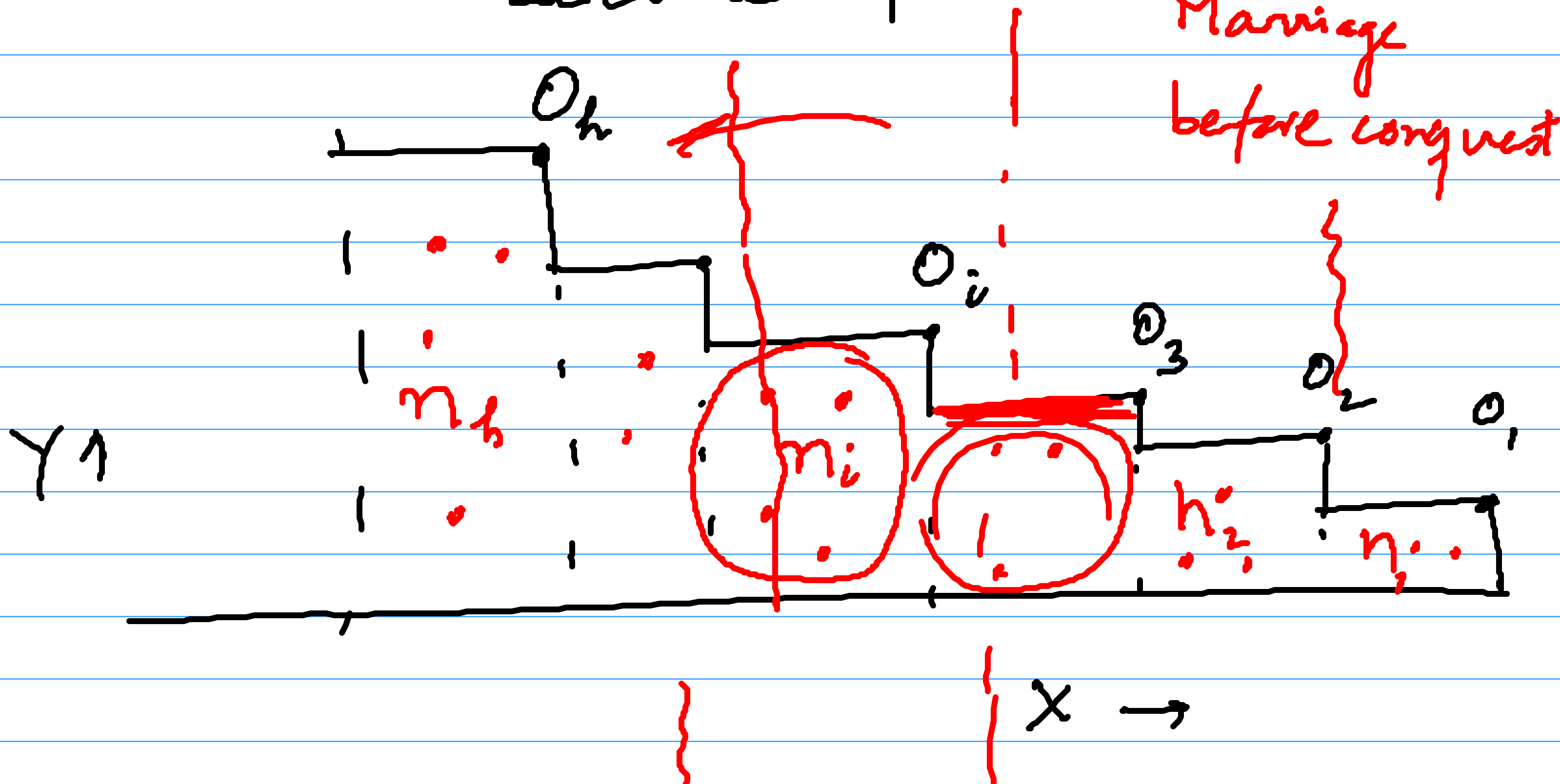
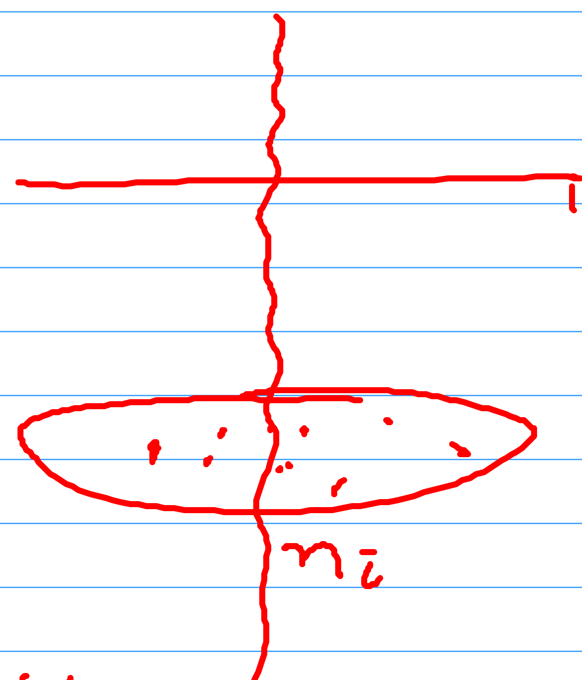
# Computational Geometry CSL 852

## Lecture 4

Marriage before conquest



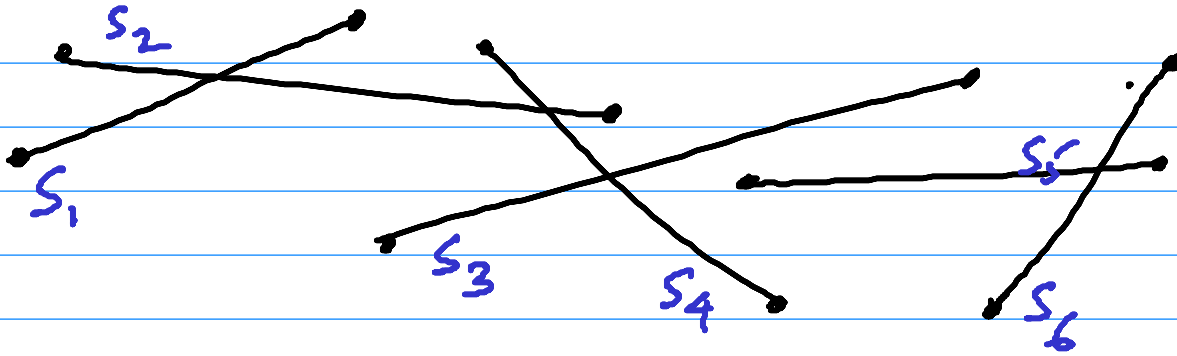

Suppose the $n_i$ points survive for $j$ levels of recursion

$\frac{n}{2^j}$ : maximum size of subproblem

$$\frac{n}{2^j} \geq n_i \Rightarrow j \leq \log\left(\frac{n}{n_i}\right)$$

Total running $\left(\sum_{i=1}^{h} n_i \times \log\left(\frac{n}{n_i}\right) \leq n \log h\right.$

# Line-Sweep method

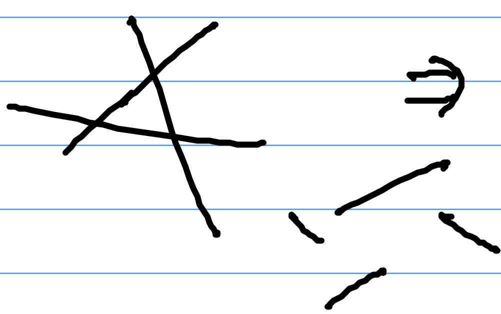



Given a set $S$ of $n$ line segments compute the intersecting pairs

$(s_1, s_2)$ $(s_2, s_4)$ $(s_3, s_4)$ $(s_5, s_6)$

- How do compute intersection between two given line segments?

- First attempt Check all $\binom{n}{2}$ pairs.



$\Rightarrow O(n^2)$ operations

# Lecture 5 Computational Geometry

Total number of event points (where the ordering $L(x)$ changes)

$= 2n$ (endpoints of $n$ segments)

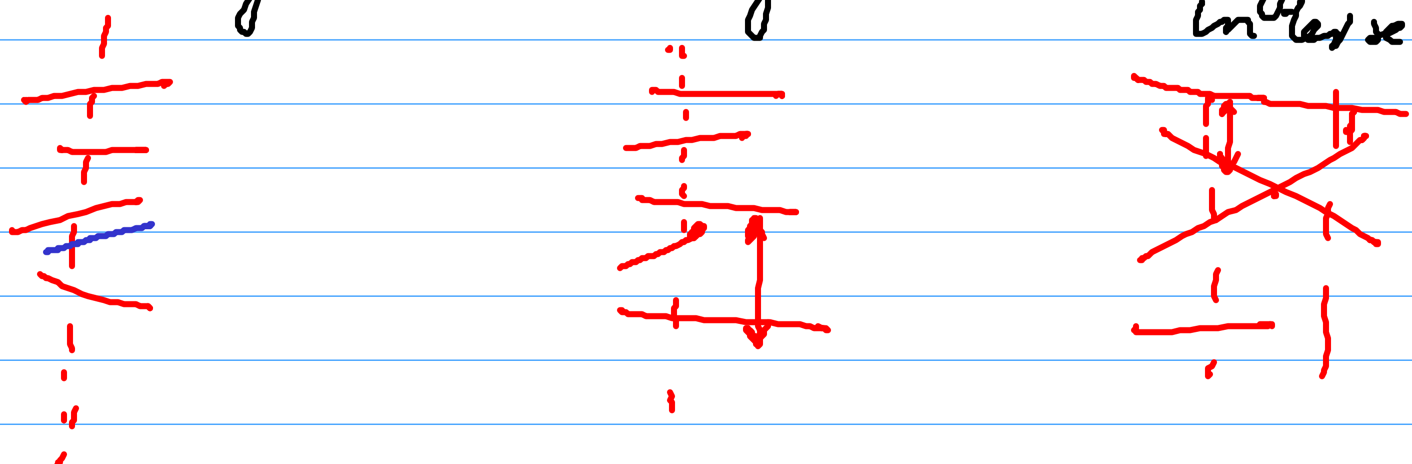
$+ I$ (# intersecting pairs)

# operations required to record the changes at each event point

Case 1 — New Segment

Case 2 — Segment ends

Case 3 — Segments intersect



Updates at any event point

- $L(x)$ changes by some constant
- For new adjacency pairs, we compute "potential intersections" and update "Event queue"

Use any efficient data structure for storing (dynamic) sorted sets like red-black trees, AVL, Skip Lists

$\Rightarrow$ $O(\log n)$ per update

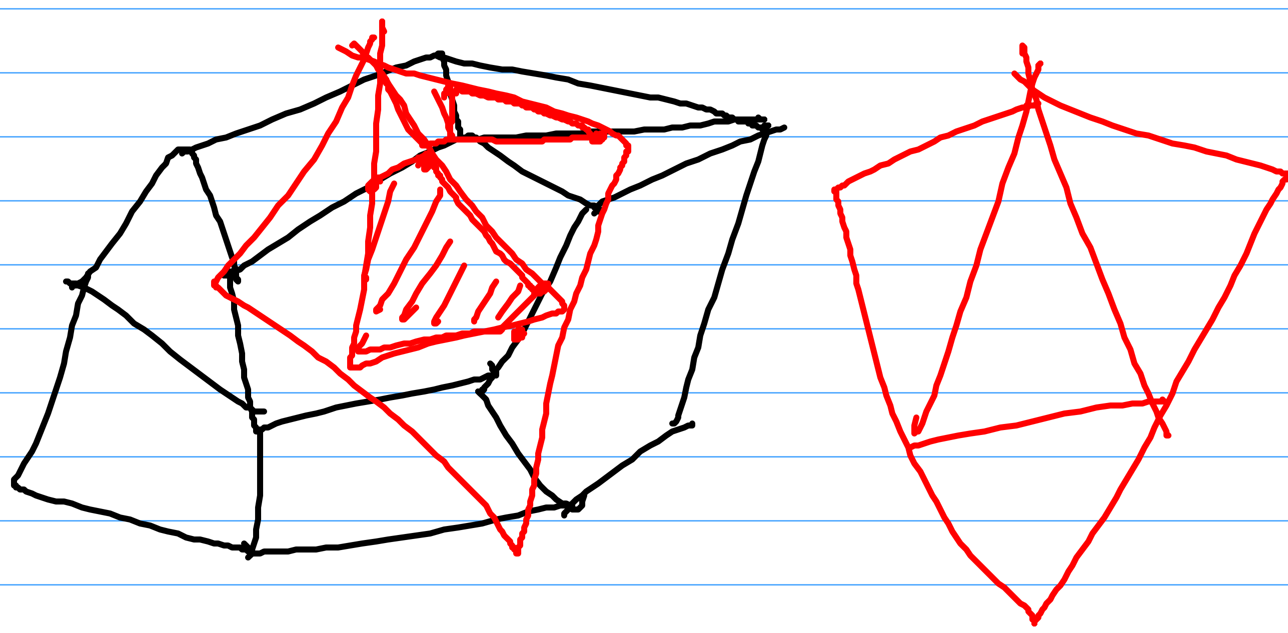
$\Rightarrow$ $O(\log n)$ operations per event point

$\Rightarrow$ $O((n+I) \cdot \log n)$ overall

$= O(\underline{n \log n} + I \log n)$

Among $n$ elements, are all of them unique? (Element Uniqueness)

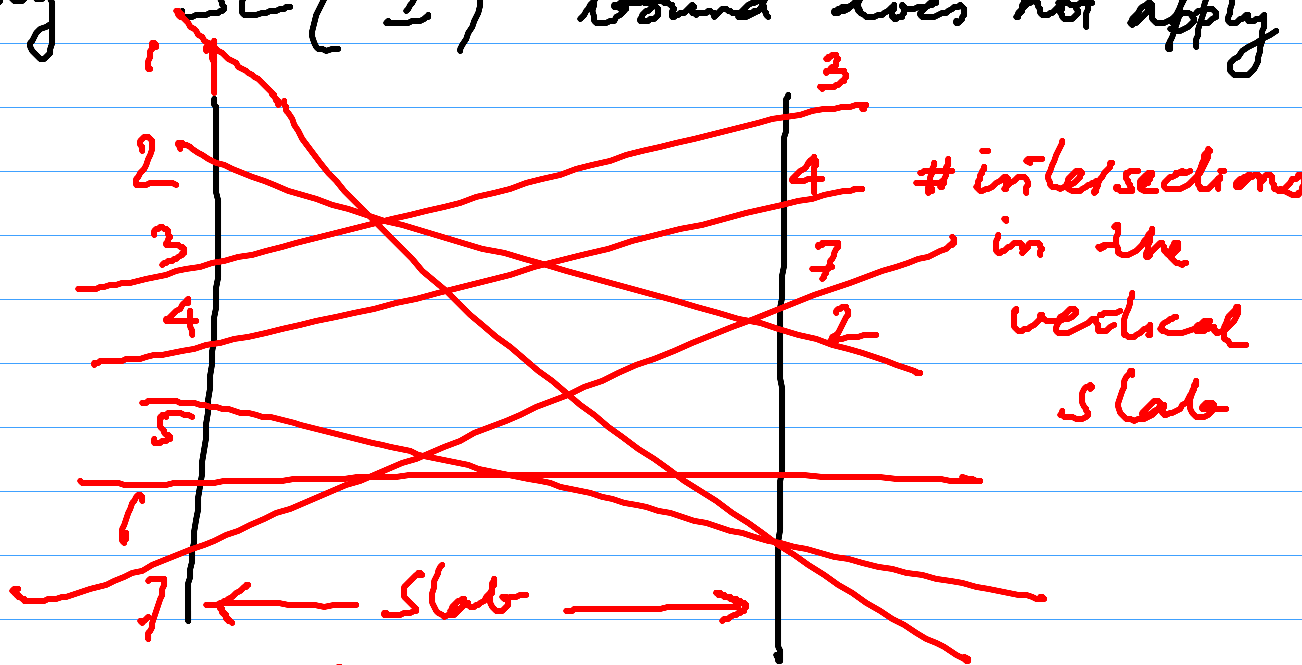
$\Omega(n \log n)$ : Linear decision tree model



## Counting Version

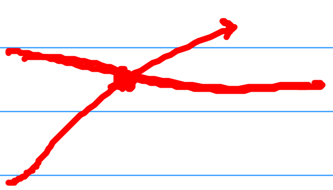
\# intersections (instead of the list of intersecting pairs)

Clearly $\Omega(I)$ bound does not apply



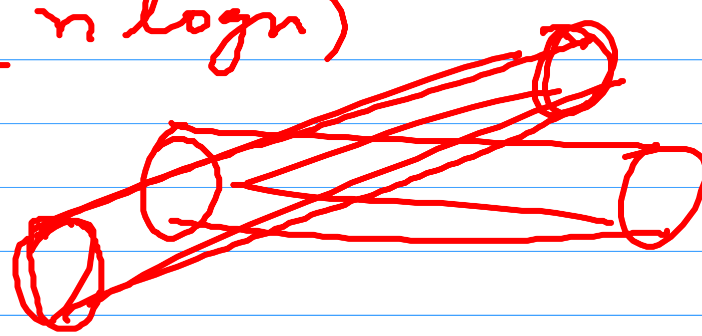

hammock

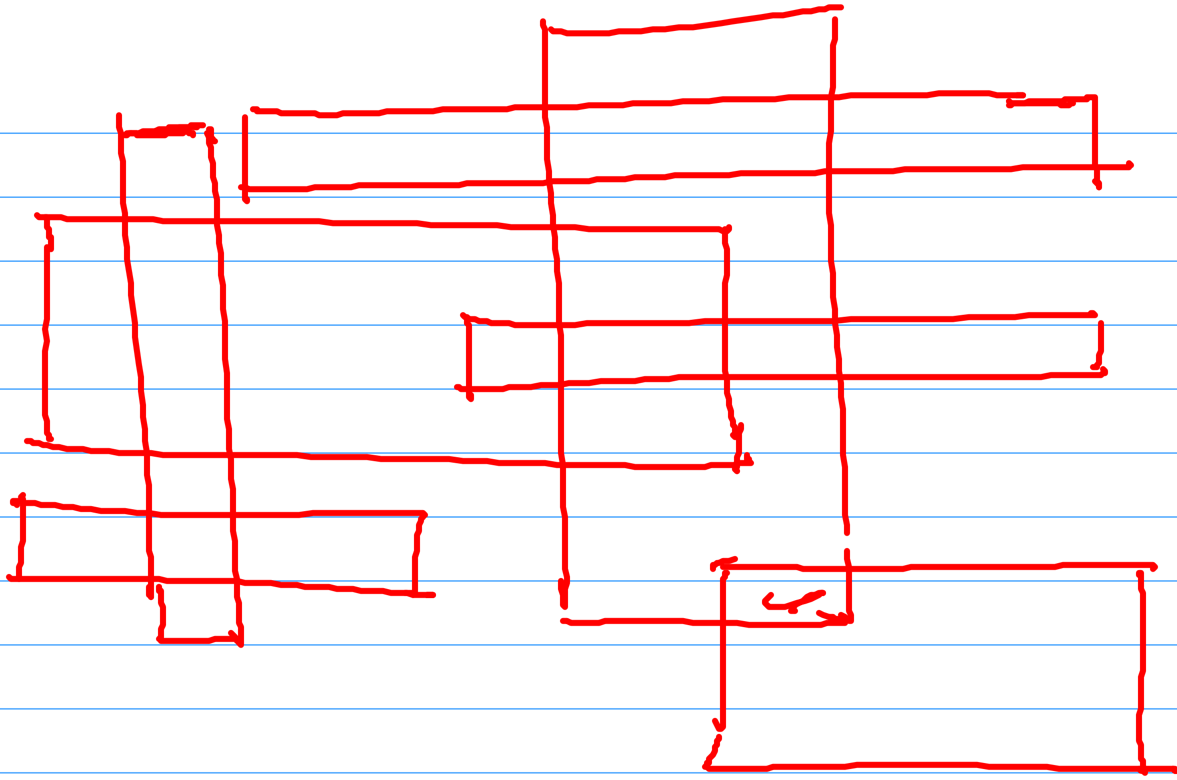
3, 1, 4, 2 : how many inversions



**Exercise:** Find # inversions $O(n \log n)$

**Arrangements:**





Area of the union of rectangles

Given set of n rectangles (possibly intersecting)