Given \( S = \{A_1, \ldots, A_n\} \), any \( n \) segments in \( \mathbb{R}^2 \)

**Get:** \( I = \{ i, j : A_i \cap A_j \neq \emptyset \} \)

\( \sigma(I) = \{ \text{distinct pts } A_i \cap A_j, 1 \leq i, j \}

\( 0 \leq |\sigma(I)| \leq |I| \leq \binom{n}{2} \)

**At first:** "gen. position"

1. No proper intersection...
2. No 3 segs at a pt.
3. No 8 verticils.

**Fix all later** (so don't restrict)

**Lower bound** \( \Omega(\log n) \) for \( |\sigma(I)| \); \( |I| \) w/ element uniqueness

We will get
1. \( O\left[ (n + |I|) \log n \right] \)
2. \( O\left[ (n + |\sigma(I)|) \log n \right] \)

Bose-Sweeney '95, \( O(n \log n + |I|) \)

**Idea:** "Sweep" vertical \( x = t \) from \(-\infty \to +\infty \)

- Discover elements of \( I \) \( (\sigma(I)) \) as sweep.

(over)
ALGORITHM FOR SEG-INT (Special Case)

- Sort \( A_i, B_i, i = 1, ..., n \) by \( P \times \left[ \begin{array}{c} x_i \\ \delta_i = \overline{A_i \cap B_i} \end{array} \right] \)

- \( Q \leftarrow \) "balanced" binary search tree for \( \delta \) endpoints

- \( t \leftarrow \min(\delta) - 1 \)

- \( T \leftarrow \emptyset \) (empty BST for current segments at \( x = t \))

- WHILE \( Q \neq \emptyset \) DO
  - get next event \( P = (x,y) \) from \( Q \) (has \( \min x \))
  - \( t \leftarrow x \) (sweep to next event \( P \))
  - \"PROCESS\( (P)\)\"
    - update \( \text{STATUS} (t) \)
    - update \( Q \) for future events caused by \( P \)
    - delete \( P \) from \( Q \)
    - balance \( Q, T \) if needed

- END WHILE

(details given)

in class