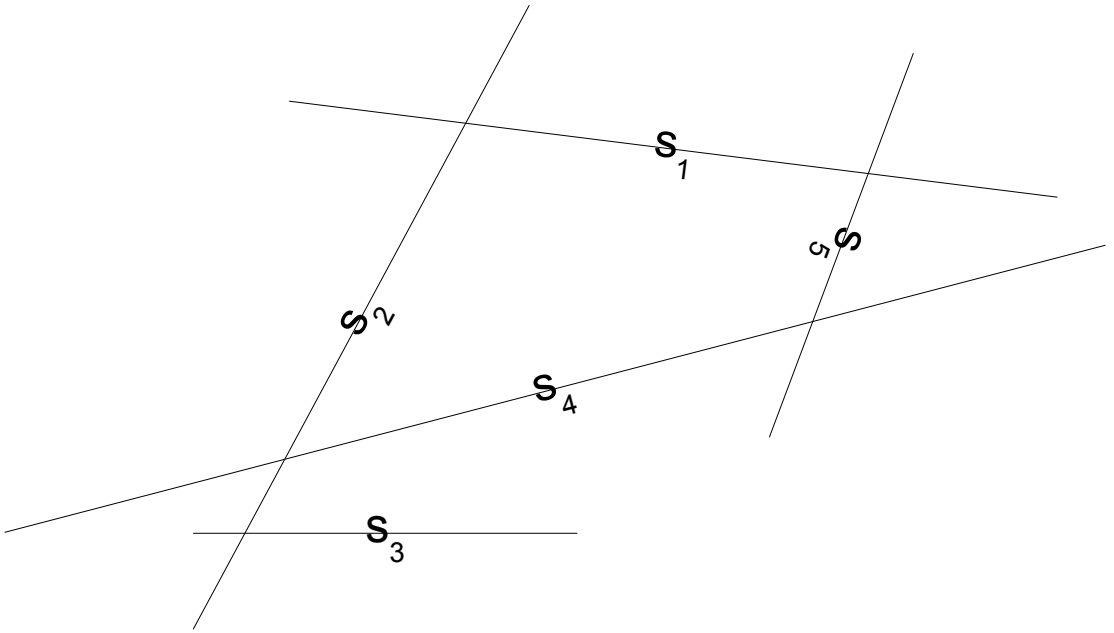
Intersection of Line Segments

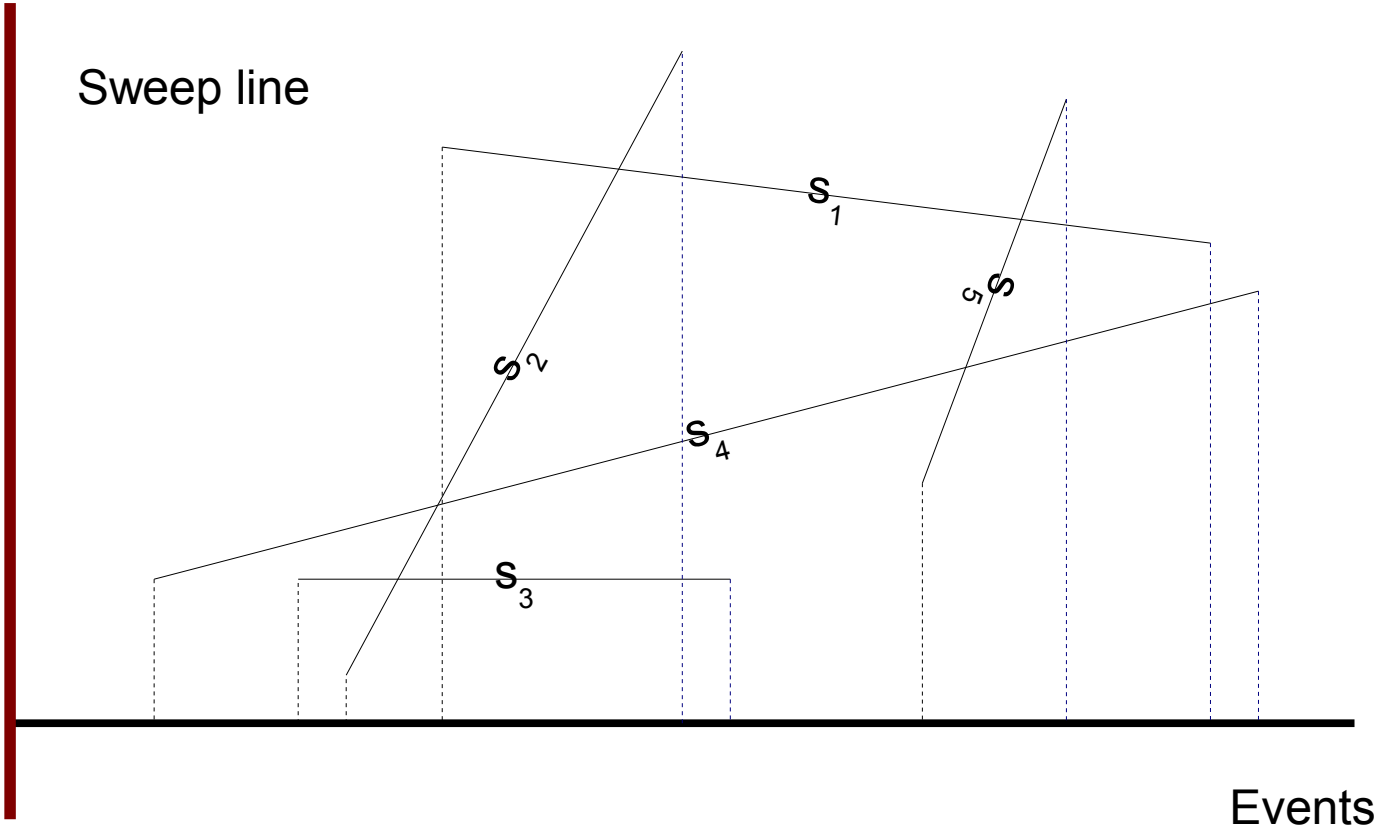
- Given: *n* line segments.
- Find: All intersections (report segments at each intersection point).



Intersection of Line Segments

- Trivial method $O(n^2)$.
- Plane sweep: $O((n+k)\log n)$ time, $O(n+k)$ space where k is the number of intersections (at most $O(n^2)$).
- Necessary condition for two lines to intersect: next to each other on some vertical line.
- Simplifying assumptions:
  - No vertical segments.
  - No segements overlap.
  - No three segments meet at the same point.

Plane Sweep Algorithm



Plane Sweep – Events

- Left endpoints of segments.
- Right endpoints of segments.
- Intersections of line segments.

Plane Sweep - Status

- Status of the sweep line at a given *x*-coordinate.
  - Segments intersected by the vertical sweep line through *x* from top to bottom.

Plane Sweep Algorithm

- Arriving at a left endpoint of a line segment *s*:
  - Insert *s* into the correct spot on the sweep line.
  - Check if *s* intersects the segment above it. If so, add the intersection to the events.
  - Check if *s* intersects the segment below it. If so, add the intersection to the events.

Plane Sweep Algorithm

- Arriving at a right endpoint of a line segment *s*.
  - Check if the segments above and below *s* intersect to the right of *s*. If so, and the intersection was not reported earlier, add it to the events.
  - Remove *s* from the status.

Plane Sweep Algorithm

- Arriving at an intersection point *p* between two line segments s_1 and s_2
  - Swap s_1 and s_2 in the status.
  - Check if s_2 and the segment above it intersect to the right of *p*. If so and *p* was not reported earlier, add it to the events.
  - Check if s_1 and the segment below it intersect to the right of *p*. If so and *p* was not reported earlier, add it to the events.

Plane Sweep Algorithm - Events

- Operations that must be supported:
  - Get the smallest (lexicographically) element and delete it.
  - Insert a new element in an appropriate position.
  - Check for membership.
- Use binary search trees.
- Every operations requires $O(\log n)$ time.

Plane Sweep Algorithm - Status

- Operations that must be supported:
  - Insert an element in an appropriate position.
  - Delete an element.
  - Swap two consecutive elements.
  - Return the element just before a given element.
  - Return the element just after a given element.
- Use binary search trees.
- Every operation requires $O(\log n)$ time.

Plane Sweep - Complexity

- Preprocessing: $O(n\log n)$ time and $O(n)$ space.
- $O((n+k)\log n)$ time:
  - Number of stops: $2n+k$.
  - Each stop requires $O(\log n)$ time.
- $O(n+k)$ space.
- $O(n)$ space: delete events if line segments seize to be neighbours on the sweep line.