Segment Intersection Sweep Line Algorithm

Main Module: Find Intersections

Algorithm: Find Intersections

Input: **S**: Set of line segments in a 2D plane.
Output: **R**: Set of intersection points within all segments of S, and for each of the points, the segments that intersect.

Initialize the Event Queue Q by inserting all end points (start and end) of all segments. When an upper end point is inserted, its corresponding segment must be stored.
Initialize the Sweep Line T, initially empty.
While Q *is not empty* **:**
    Obtain and Delete the next Event p from Q.
    $processEvent(p)$
Return R

Module: Process Event

Algorithm: Process Event

Input: Event p

Let $U(p)$ be the set of segments from S that have their upper end point at p. In case of a horizontal segment the upper end point is the left-most point.
Find all segments in T that contain p; they must be adjacent in T. Let $L(p)$ be a sub-set of segments found whose lower end point is p, and $C(p)$ a sub-set of segments that contain p within themselves.
if $|L \cup U \cup C| > 1$ **then**
    Report p as an intersection point with all segments of L, U, C.
end
Delete the segments $L \cup C$ from T.
Insert the segments $U \cup C$ in T. The order in T must correspond to the order in which the segments intersect the Sweep Line just below p. If there is a horizontal segment, insert it at the end.
if $U \cup C = \emptyset$ **then**
    Let s_l and s_r the left and right neighbours of p over T, $findEvents(s_l, s_p, p)$.
end
else
    Let s' be the left-most segment of $U \cup C$ in T.
    Let s_l be the left neighbour of s' in T.
    $findEvents(s_l, s', p)$.
    Let s'' be the right-most segment of $U \cup C$ in T.
    Let s_r be the right neighbour of s'' in T.
    $findEvents(s'', s_p, p)$.
end

Module: Find Events

Algorithm: Find Events

Input: s_l, s_r, p

---

1 **if** *s_l and s_r intersect under the Sweep Line, or right at it but at the left side of current Event p, and the intersection is not yet an Event in Q* **then**
2 | **Insert** the intersection point as a new event in Q.
3 **end**