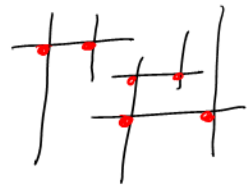
An Application of Red Black Tree

Input: A set S of n-line segments consisting of Horizontal and Vertical Segments.

Output: Report all intersections.



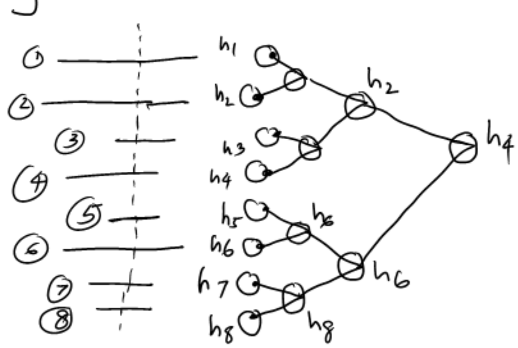
ALGORITHM

Step1: Sort the endpoints of all horizontal segments in ascending order (w.r.t x-coordinate)
Let H be the sorted list of these endpoints.

Step2: Sort the vertical segments w.r.t to x-coordinate in ascending order.
Let V be the sorted list of vertical segments.

Step3: Merge the list $H \cup V$.

Step4: Horizontal segments will be maintained in a red-black tree T, where the key is the y-coordinate.



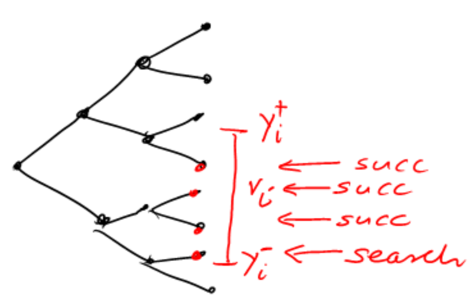
For each vertical segment v, T is searched to find all horizontal segments intersecting v.

<u>Step5</u>: Elements will be considered one by one from HUV and one of the following actions will be performed:

<u>Case 1</u>: If the next element is a left end point of horizontal segment h_i then insert h_i in T, with its key as the y-coordinate of h_i.

<u>Case 2</u>: If the next element is a right end point of horizontal segment h_i then delete h_i from T.

<u>Case 3</u>: If the next element is a vertical segment $v_i = (y_i^-, y_i^+)$ then search in T to locate y_i^-. Keep reporting the successor element till the key is $\leq y_i^+$.



Analysis

- Forming lists H U V costs $O(n\log n)$ time.
  (sorting n-elements w.r.t x-coordinate).

- Insertion of each left endpoint of a horizontal segment
  Deletion of each right endpoint of a horizontal segment
  costs $O(\log n)$ time.

  Total Insertion / Deletion cost is
  $O(|H| \log n)$.

- For each vertical segment we perform
  - one search $O(\log n)$
  - # succ computations = # intersections.

- Let K = Total # intersections between horizontal and vertical segments.

- Total cost for vertical segments

$$= O\Big(\underbrace{|V| \log n}_{\text{cost for search}} + \underbrace{K \cdot \log n}_{\text{cost for succ.}} \Big)$$

- Total cost for computing intersections

$$= O\big(|H| \log n + |V| \log n + K \cdot \log n \big)$$

$$= O\big(n \log n + K \log n \big).$$

$\boxtimes$