Problem：給定 $S_1, S_2, \ldots, S_n$ 為 n 个 line segment 之 endpoint
判斷 $S_1, \ldots, S_n$ 中是否存在某一 pair 之 line segment 有相交

brute force：兩兩執行判斷兩 line segment 有無相交之 Algorithm：$O(1) \cdot O(C_2^n) = O(n^2)$

Sweeping technique：考慮一條和 x 軸垂直之 sweep line，由左至右做 sweeping
在掃到每个 event point 時，更新 sweep line statue，直至 Algorithm 終止

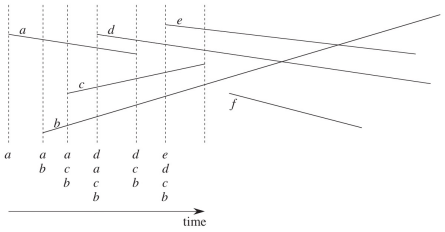


Figure 33.5 The execution of ANY-SEGMENTS-INTERSECT. Each dashed line is the sweep line at an event point. Except for the rightmost sweep line, the ordering of segment names below each sweep line corresponds to the total preorder T at the end of the **for** loop processing the corresponding event point. The rightmost sweep line occurs when processing the right endpoint of segment c; because segments d and b surround c and intersect each other, the procedure returns TRUE.

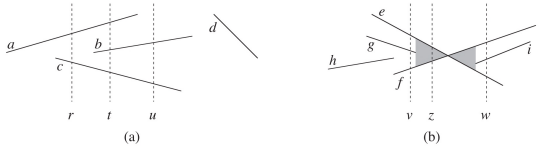


Figure 33.4 The ordering among line segments at various vertical sweep lines. **(a)** We have $a \succcurlyeq_r c$, $a \succcurlyeq_t b$, $b \succcurlyeq_t c$, $a \succcurlyeq_t c$, and $b \succcurlyeq_u c$. Segment d is comparable with no other segment shown. **(b)** When segments e and f intersect, they reverse their orders: we have $e \succcurlyeq_v f$ but $f \succcurlyeq_w e$. Any sweep line (such as z) that passes through the shaded region has e and f consecutive in the ordering given by the relation $\succcurlyeq_z$.

Event points：給定 n 个 line segment，將所有 line segment 之 end-point 用 x 坐標由小至大排序
其中：若 x 坐標相同，以 y 坐標小的優先
且所有 left endpoint 排在 right endpoint 之前

Sweepline statue：當 sweep line 停在某 event point 時，把所有和 sweep line 相交之 line segment
以 y 坐標小至大排序，得 total preorder T

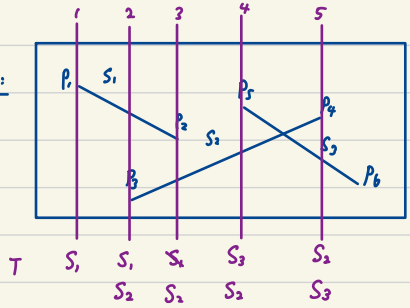
Algorithm:

1. 將 event point 依 (x_i, y_i) 坐標排序
2. 依 sorting 順序檢查每個 event point p
3. 設 p 為 S_i 之 endpoint，而 S_1, S_2 為 T 中和 S_i 相鄰之2個 line segment
   (1). 若 p 為 S_i 之 left endpoint，則將 S_i 加入 T
   若 S_i 和 S_1 或 S_2 相交，return True
   (2). 否則，p 為 S_i 之 right endpoint
   若 S_i 和 S_1 或 S_2 相交，return True
   將 S_i 從 T 刪除
4. 若檢查完每個 event point，則 return False

```
ANY-SEGMENTS-INTERSECT(S)
1  T = ∅
2  sort the endpoints of the segments in S from left to right,
       breaking ties by putting left endpoints before right endpoints
       and breaking further ties by putting points with lower
       y-coordinates first
3  for each point p in the sorted list of endpoints
4      if p is the left endpoint of a segment s
5          INSERT(T, s)
6          if (ABOVE(T, s) exists and intersects s)
               or (BELOW(T, s) exists and intersects s)
7              return TRUE
8      if p is the right endpoint of a segment s
9          if both ABOVE(T, s) and BELOW(T, s) exist
               and ABOVE(T, s) intersects BELOW(T, s)
10             return TRUE
11         DELETE(T, s)
12 return FALSE
```

- INSERT(T, s): insert segment s into T.
- DELETE(T, s): delete segment s from T.
- ABOVE(T, s): return the segment immediately above segment s in T.
- BELOW(T, s): return the segment immediately below segment s in T.

Example:



① 依 event point sorting: $P_1, P_3, P_2, P_5, P_4, P_6$

② 由左至右掃過

∵在 sweep line 4, 5 時，在 T 中 $S_3 \le_4 S_2$ 變為 $S_2 \le_5 S_3$

∴ 可知 S_2, S_3 有相交

Time Complexity:

① 將 endpoint 排序：$O(n \lg n)$

② 用紅黑樹 implement T，∴ insert, delete, above, below 皆為 $O(\lg n)$

⇒ $T(n) = O(n \lg n)$