

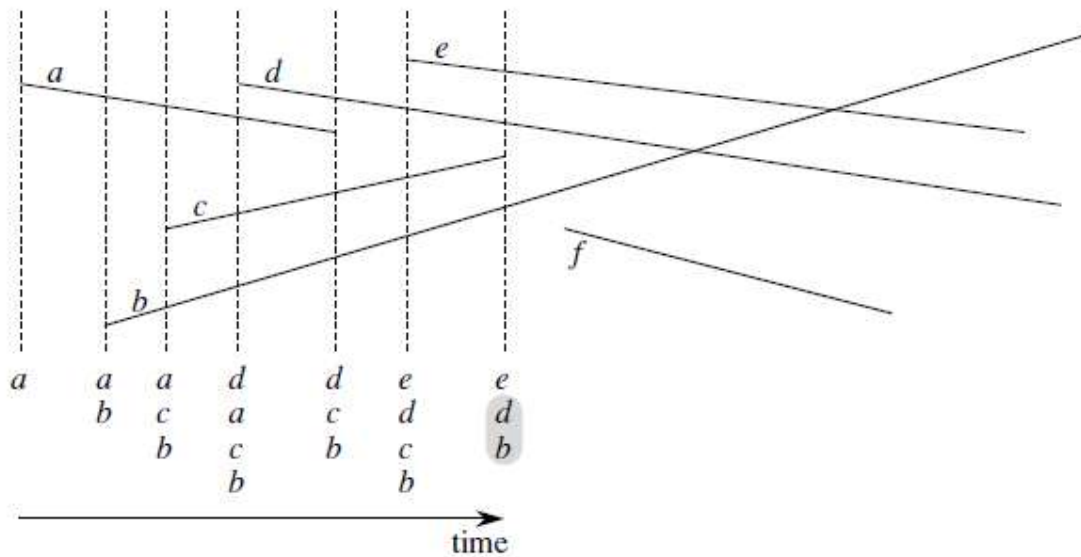
Determining whether any pair of segments intersects

(Sweeping Lines)

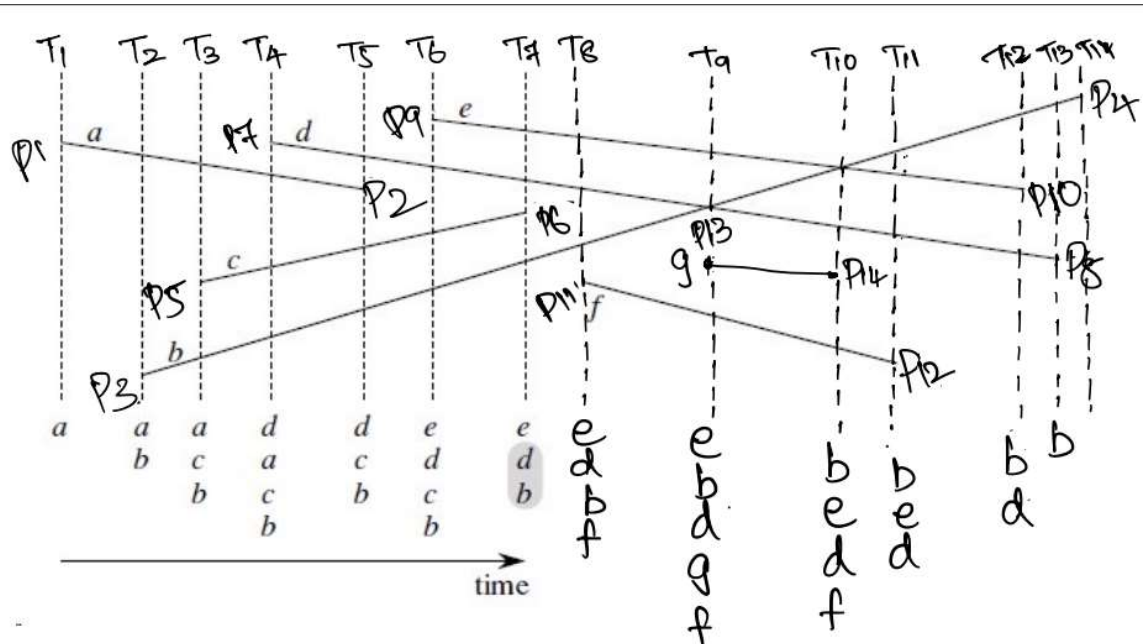
- An algorithm for determining **whether any two line segments in a set of segments intersect** uses a technique known as “**sweeping**,” which is common to many computational-geometry algorithms.
- The algorithm runs in $O(n \lg n)$ time, where n is the number of segments we are given. **It determines only whether or not any intersection exists; it does not print all the intersections.**
- In **sweeping**, an imaginary vertical **sweep line** passes through the set of geometric objects (i.e., set of points), usually from left to right.
- The spatial dimension through which the sweep line moves across is treated as dimension of time. In this case of determining whether any two line segments in a set of segments intersect, the x-dimension, is treated as a dimension of time.
- Sweeping provides a method for ordering geometric objects.
- **Line-segment-intersection algorithm** that has been discussed already is used here to check for an intersection.
- Algorithm for determining whether any two of n line segments intersect, makes **two simplifying assumptions**.
 - First, we assume that no input segment is vertical.
 - Second, we assume that no three input segments intersect at a single point i.e., only 2 segments intersect.

Ordering segments

- We can order the segments that intersect a vertical sweep line according to the y-coordinates of the points of intersection.



- What happens when the sweep line passes through the intersection of two segments?
 - Their positions in the total order are reversed. Check total order at time *T*₈ and *T*₉ and *T*₁₀.



- Sweep line T_8 is to the left of the intersection point of segments d and b , and we have $d >_{T_8} b$. But at sweep line T_9 which passes through intersection point, their positions in the total order is reversed i.e. we have $b >_{T_9} d$.
 - In the same way sweep line T_9 is to the left of the intersection point of segments e and b , and we have $e >_{T_9} b$. But at sweep line T_{10} which passes through intersection point, their positions in the total order is reversed i.e. we have $b >_{T_{10}} e$.
 - Note that **there must be some vertical sweep line T_x for which intersecting segments are *consecutive* in the total order**. Check sweep line at time T_7 . Segments d and b are consecutive.
- Sweeping algorithm typically manage two sets of data:
 - The ***sweep-line status*** is a total order T . We order the segments that intersect a vertical sweep line according to the y -coordinates of the points of intersection.

The sweep-line status is a total order T , for which we require the following operations:

 - $\text{INSERT}(T, s)$: insert segment s into T
 - $\text{DELETE}(T, s)$: delete segment s from T
 - $\text{ABOVE}(T, s)$: return the segment immediately above segment s in T
 - $\text{BELOW}(T, s)$: return the segment immediately below segment s in T - The ***event-point schedule*** is a sequence of x -coordinates, ordered from left to right. Changes to the sweep-line status occur only at event points.

Segment-intersection pseudocode

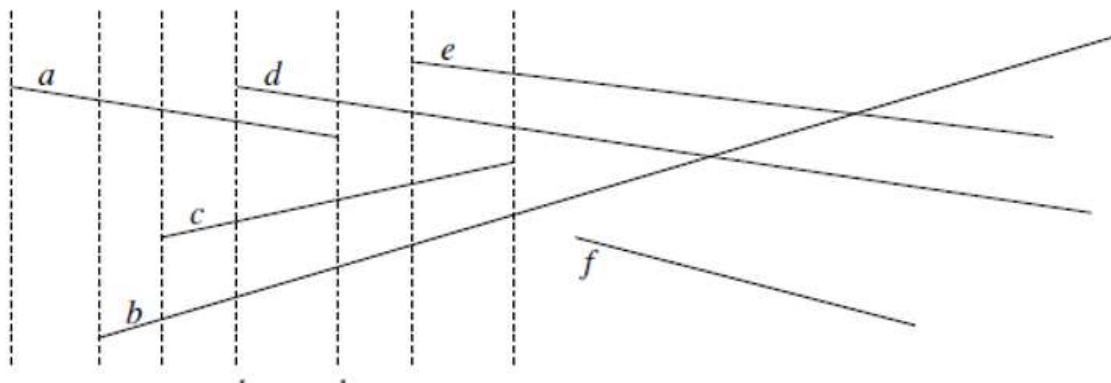
The following algorithm takes as input a set S of n line segments, returning the boolean value TRUE if any pair of segments in S intersects, and FALSE otherwise.

ANY-SEGMENTS-INTERSECT(S)

```
1   $T \leftarrow \emptyset$ 
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      do if  $p$  is the left endpoint of a segment  $s$ 
5          then INSERT( $T, s$ )
6              if (ABOVE( $T, s$ ) exists and intersects  $s$ )
                  or (BELOW( $T, s$ ) exists and intersects  $s$ )
7                  then return TRUE
8      if  $p$  is the right endpoint of a segment  $s$ 
9          then if both ABOVE( $T, s$ ) and BELOW( $T, s$ ) exist
                  and ABOVE( $T, s$ ) intersects BELOW( $T, s$ )
10                 then return TRUE
11                 DELETE( $T, s$ )
12 return FALSE
```

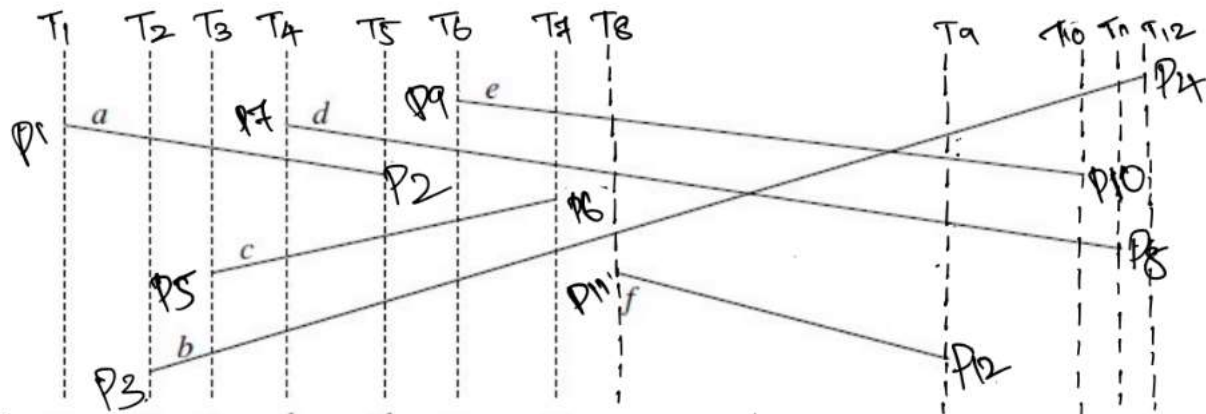
Trace out of Pseudocode

Let us consider segment a, b, c, d, e and f.



Let $a = p_1p_2$, $b = p_3p_4$, $c = p_5p_6$, $d = p_7p_8$, $e = p_9p_{10}$, $f = p_{11}p_{12}$

Draw sweep lines at each point from left to right along x-axis. Let us label sweep lines as T_x i.e., $T_1, T_2, \dots$



$$T = \emptyset$$

event pts sort = $\{ p_1, p_3, p_5, p_7, p_2, p_9, p_6, p_{11}, p_{12}, p_{10}, p_8, p_4 \}$
(event Pt. schedule)

For loop begins.

① $p_1 \rightarrow$ left end pt. segment $s = a$.

$$T = a$$

If (Above(T, s) exists & intersects s) $\Rightarrow F$

$\downarrow$
 F

Below (T, s) exists & intersects s) $\Rightarrow F$

$\downarrow$
 F

② P3 → left end Pt. segment $s = b$.

if (A(T, s) exists as intersects s) ⇒ F
 ↓ check intersection of a and b
 T = a
 b
 ↓
 A(T, s) = a

u j k
 $P_3 P_4 \leftarrow P_1 \Rightarrow d_1 < 0$
 $P_3 P_4 \leftarrow P_2 \Rightarrow d_2 < 0$
 $P_1 P_2 \leftarrow P_3 \Rightarrow d_3 > 0$
 $P_1 P_2 \leftarrow P_4 \Rightarrow d_4 < 0$

do not intersect.
 ∴ false.
 do not intersect.

if (B(T, s) exists as intersects s) ⇒ F
 ↓
 F

③ P5 → left end. Pt. segment $s = c$.

if (A(T, s) exist as intersects s) ⇒ F.
 ↓ check intersection of a and c
 T = a
 c
 b
 ↓
 A(T, s) = a

u j k
 $P_5 P_6 \leftarrow P_1 \Rightarrow d_1 < 0$
 $P_5 P_6 \leftarrow P_2 \Rightarrow d_2 < 0$
 $P_1 P_2 \leftarrow P_5 \Rightarrow d_3 > 0$
 $P_1 P_2 \leftarrow P_6 \Rightarrow d_4 = 0$

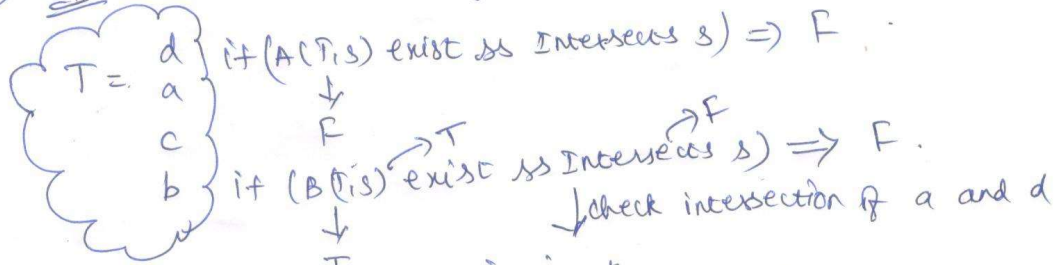
do not intersect.
 ∴ false.
 check on-segment.
 (or) on the line passing through
 returns false.

if (B(T, s) exist as intersects s) ⇒ F
 ↓ checks intersection of b and c.
 T
 B(T, s) = b

u j k
 $P_5 P_6 \leftarrow P_3 \Rightarrow d_1 > 0$
 $P_5 P_6 \leftarrow P_4 \Rightarrow d_2 < 0$
 $P_3 P_4 \leftarrow P_5 \Rightarrow d_3 < 0$
 $P_3 P_4 \leftarrow P_6 \Rightarrow d_4 < 0$

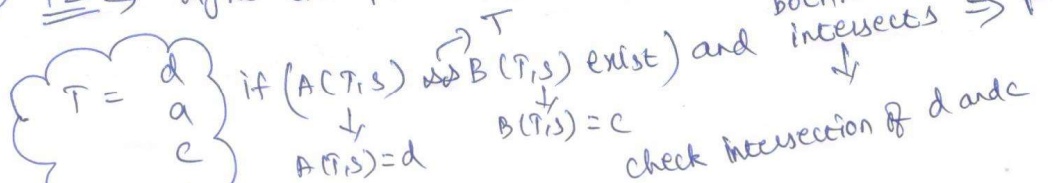
∴ false
 do not intersect

④ P7 → left end pt. segment $s = d$.



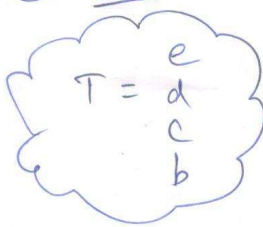
$$\begin{array}{l}
 i \quad j \quad k \\
 P7 \quad P8 \leftarrow P1 \Rightarrow d1 > 0 \\
 P7 \quad P8 \leftarrow P2 \Rightarrow d2 > 0 \\
 P1 \quad P2 \leftarrow P7 \Rightarrow d3 < 0 \\
 P7 \quad P8 \leftarrow P8 \Rightarrow d4 < 0
 \end{array}
 \left. \vphantom{\begin{array}{l} P7 \quad P8 \leftarrow P1 \Rightarrow d1 > 0 \\ P7 \quad P8 \leftarrow P2 \Rightarrow d2 > 0 \\ P1 \quad P2 \leftarrow P7 \Rightarrow d3 < 0 \\ P7 \quad P8 \leftarrow P8 \Rightarrow d4 < 0 \end{array}} \right\} \begin{array}{l} \therefore \text{false} \\ \text{do not intersect.} \end{array}$$

⑤ P2 → right end pt. segment $s = a$.



$$\begin{array}{l}
 i \quad j \quad k \\
 P5 \quad P6 \leftarrow P7 \Rightarrow d1 < 0 \\
 P5 \quad P6 \leftarrow P8 \Rightarrow d2 > 0 \\
 P7 \quad P8 \leftarrow P5 \Rightarrow d3 > 0 \\
 P7 \quad P8 \leftarrow P6 \Rightarrow d4 > 0
 \end{array}
 \left. \vphantom{\begin{array}{l} P5 \quad P6 \leftarrow P7 \Rightarrow d1 < 0 \\ P5 \quad P6 \leftarrow P8 \Rightarrow d2 > 0 \\ P7 \quad P8 \leftarrow P5 \Rightarrow d3 > 0 \\ P7 \quad P8 \leftarrow P6 \Rightarrow d4 > 0 \end{array}} \right\} \begin{array}{l} \text{false} \\ \text{do not intersect.} \end{array}$$

⑥ P₉ → left end pt. segment $s = e$.



if $(A(T, s) \xrightarrow{T} \text{exist and intersects } s) \Rightarrow F$.

if $(B(T, s) \xrightarrow{T} \text{exist and intersects } s) \Rightarrow F$.

$\downarrow$
 $B(T, s) = d$

$\downarrow$
check intersection of d and e

$i \quad j \quad k$

$P_9 \ P_{10} \ P_7 \Rightarrow d_1 > 0$

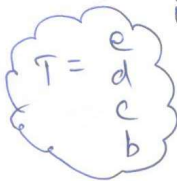
$P_9 \ P_{10} \ P_8 \Rightarrow d_2 > 0$

$P_7 \ P_8 \ P_9 \Rightarrow d_3 \leq 0$

$P_7 \ P_8 \ P_{10} \Rightarrow d_4 < 0$

False
do not intersect

⑦ P₆ → right end pt. segment $s = c$.



if $(A(T, s) \xrightarrow{T} \text{exist})$ and $(B(T, s) \xrightarrow{T} \text{exist})$ and both intersect $s \Rightarrow$ True $\Rightarrow d$ and b intersect

$\downarrow$
 $A(T, s) = d$

$\downarrow$
 $B(T, s) = b$

$\downarrow$
check intersection of d and b

$i \quad j \quad k$

$P_3 \ P_4 \leftarrow P_7 \Rightarrow d_1 < 0$

$P_3 \ P_4 \leftarrow P_8 \Rightarrow d_2 > 0$

$P_7 \ P_8 \leftarrow P_3 \Rightarrow d_3 > 0$

$P_7 \ P_8 \leftarrow P_4 \Rightarrow d_4 < 0$

True
intersects.

Therefore, algorithm returns true which means pair of segments in S intersects.