# CPSC 490 – Problem Solving in Computer Science

## Lecture 18: Line Sweep I

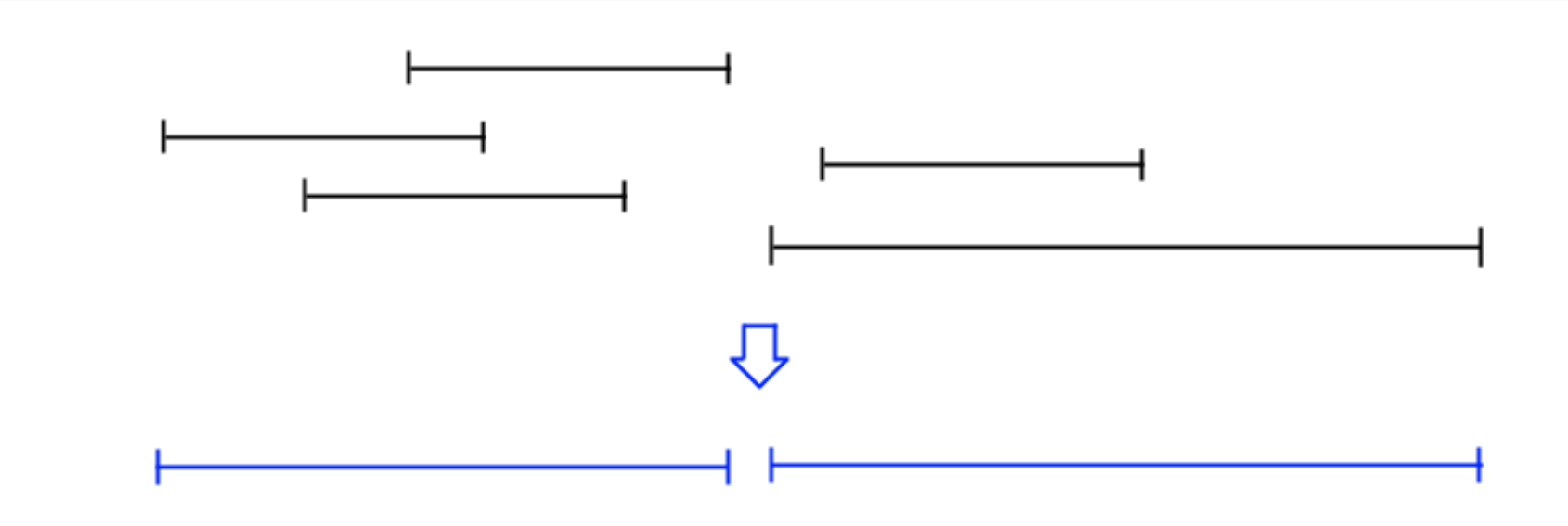
Jason Chiu and Raunak Kumar

2017/03/06

University of British Columbia

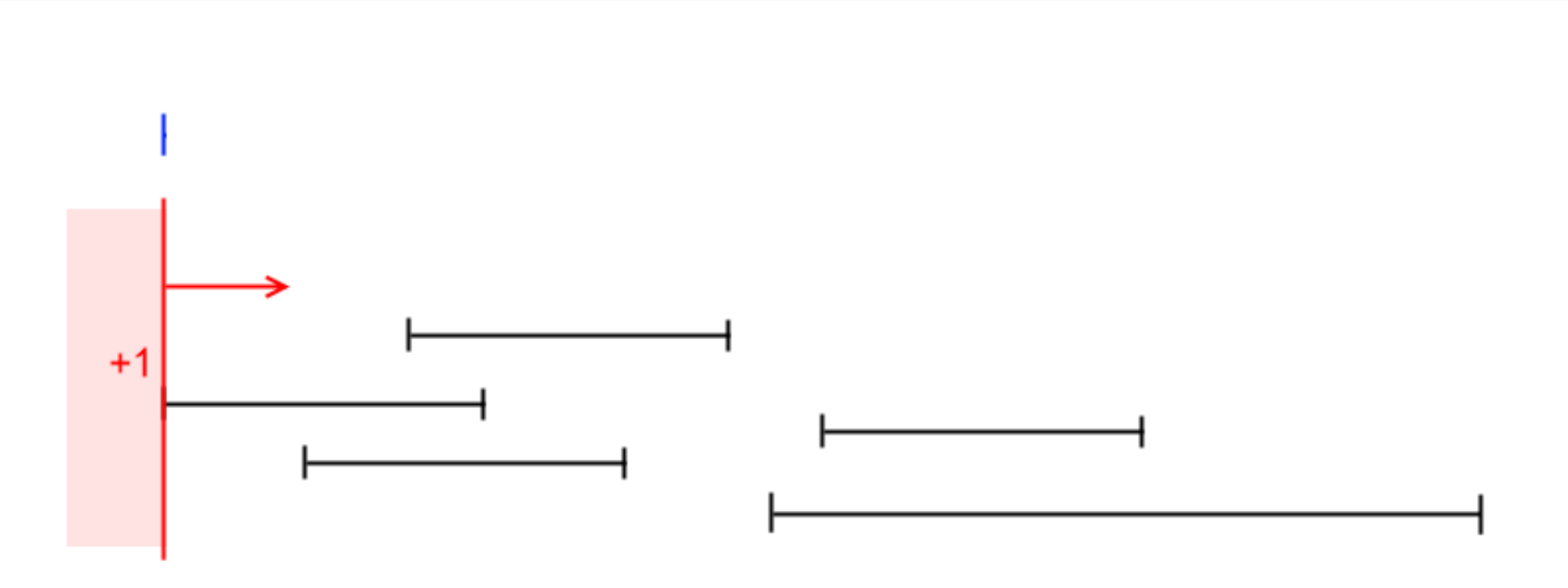
# Warm-up: Interval Union in 1D

Merge these overlapping intervals into bigger, non-overlapping intervals!



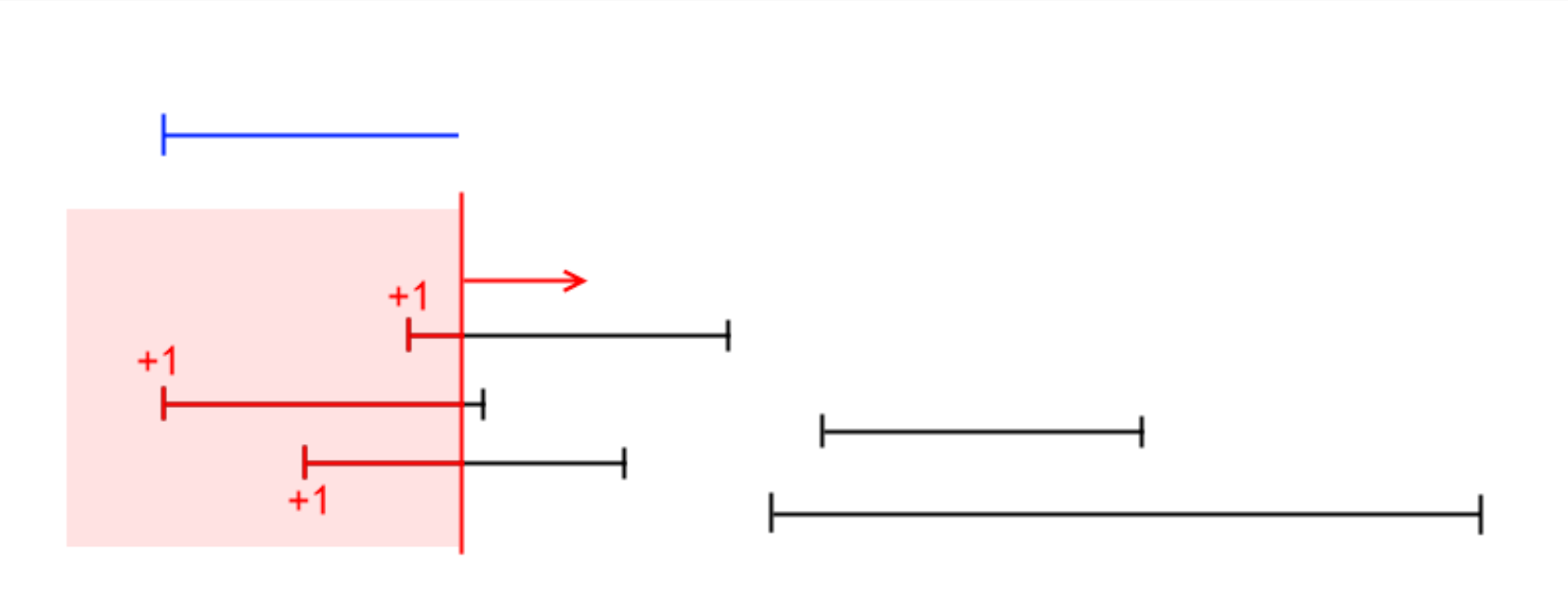
# Warm-up: Solution – 1D Line Sweep

Imagine sweeping a vertical line from left to right.



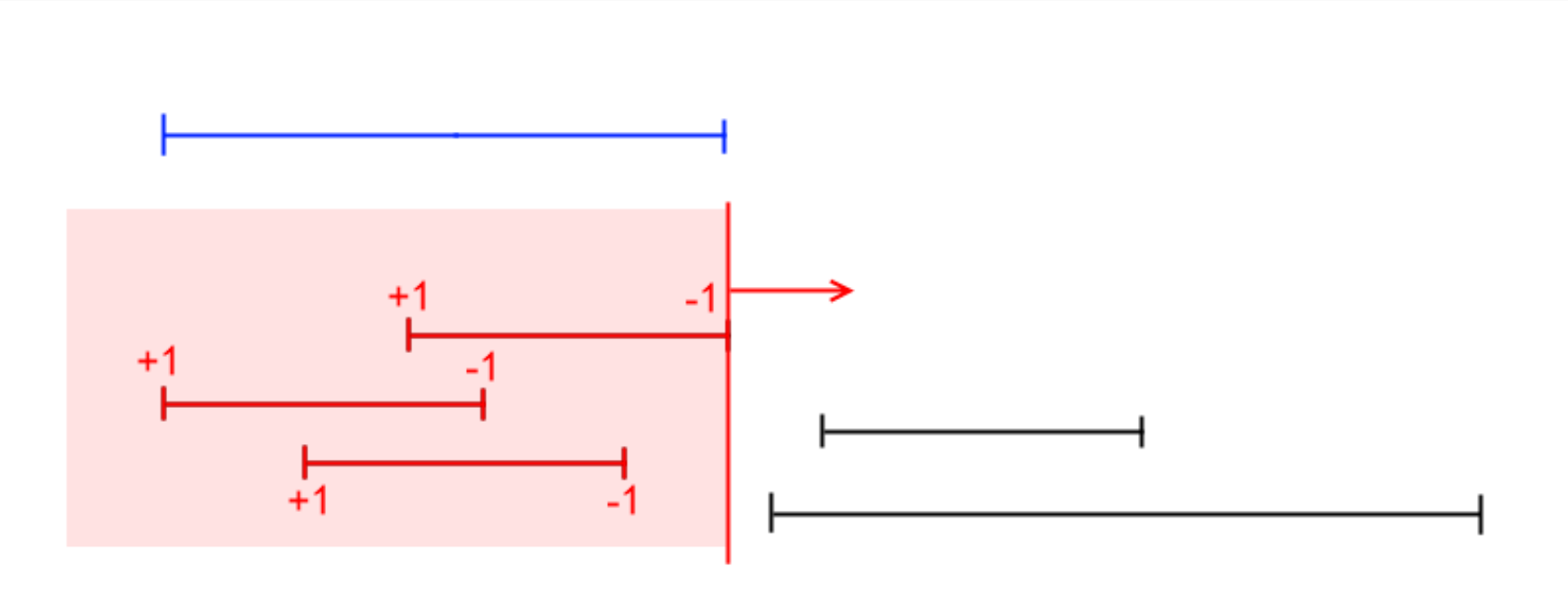
# Warm-up: Solution – 1D Line Sweep

Imagine sweeping a vertical line from left to right.



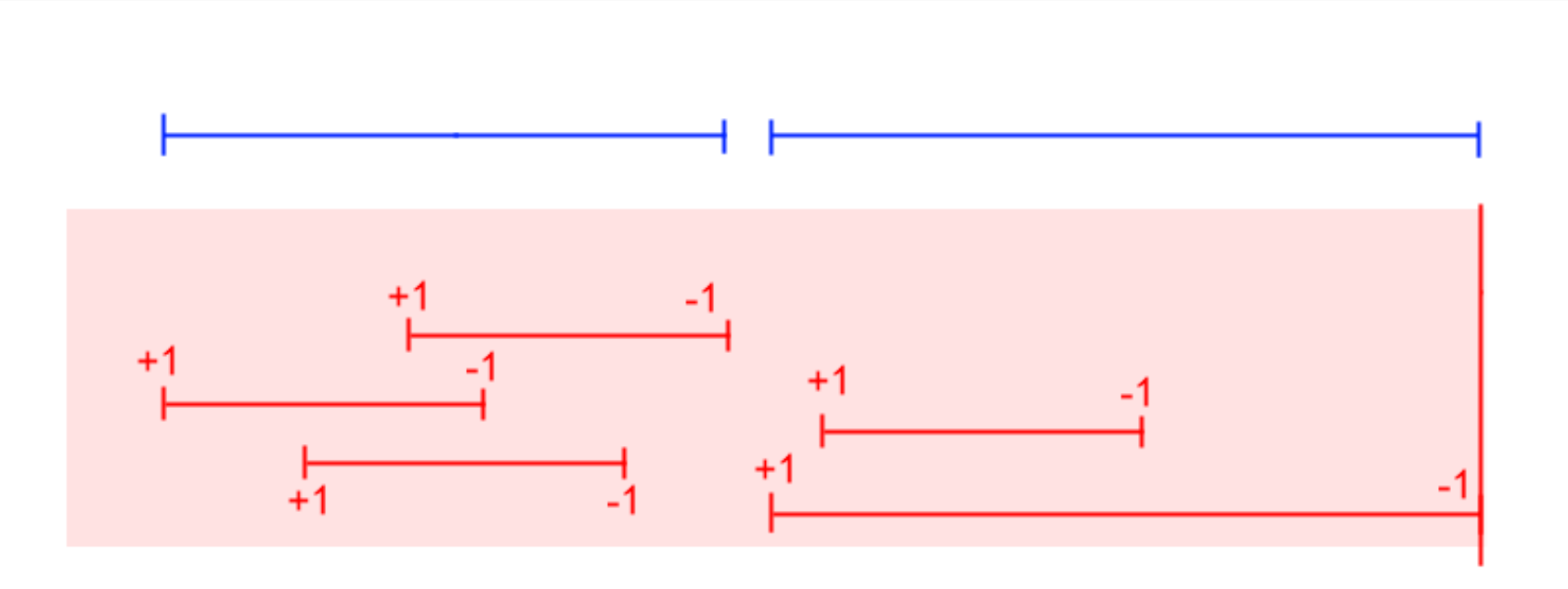
# Warm-up: Solution – 1D Line Sweep

Imagine sweeping a vertical line from left to right.



# Warm-up: Solution – 1D Line Sweep

Imagine sweeping a vertical line from left to right.



# Warm-up: Solution – 1D Line Sweep

Imagine sweeping a vertical line from left to right.

State: how many intervals sweep line intersects

Events: start and end of an interval

Actions:

- State change from 0 to positive $\rightarrow$ start an interval
- State changes from positive to 0 $\rightarrow$ end the current interval

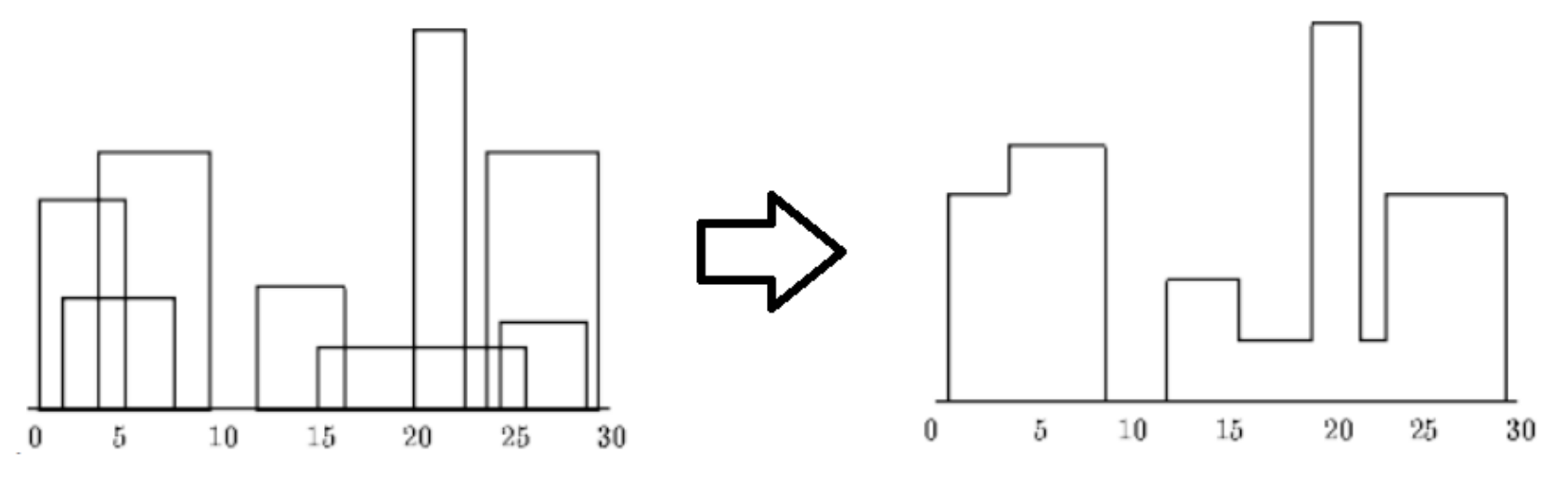
Time complexity: $O(n \log n)$ because sorting $n$ events

Be careful handling multiple events at same position!

In this problem, you must handle all events at the same position before doing any action.

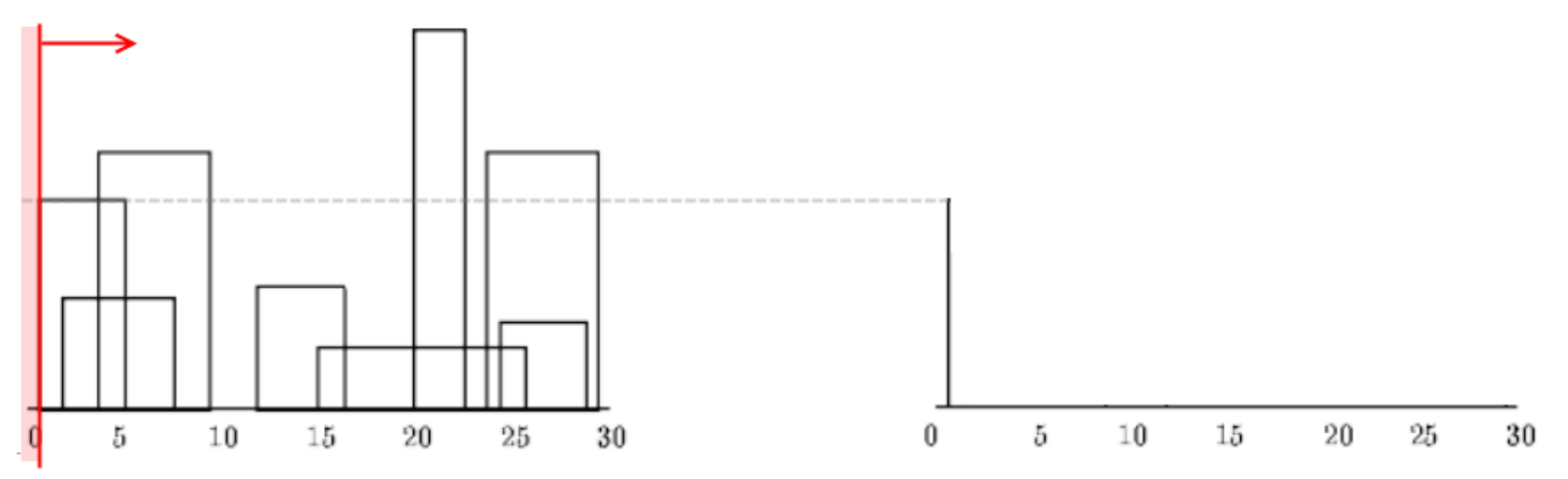
# Problem 1: Skyline

Given some rectangle sitting on $x$-axis, draw their “skyline”!



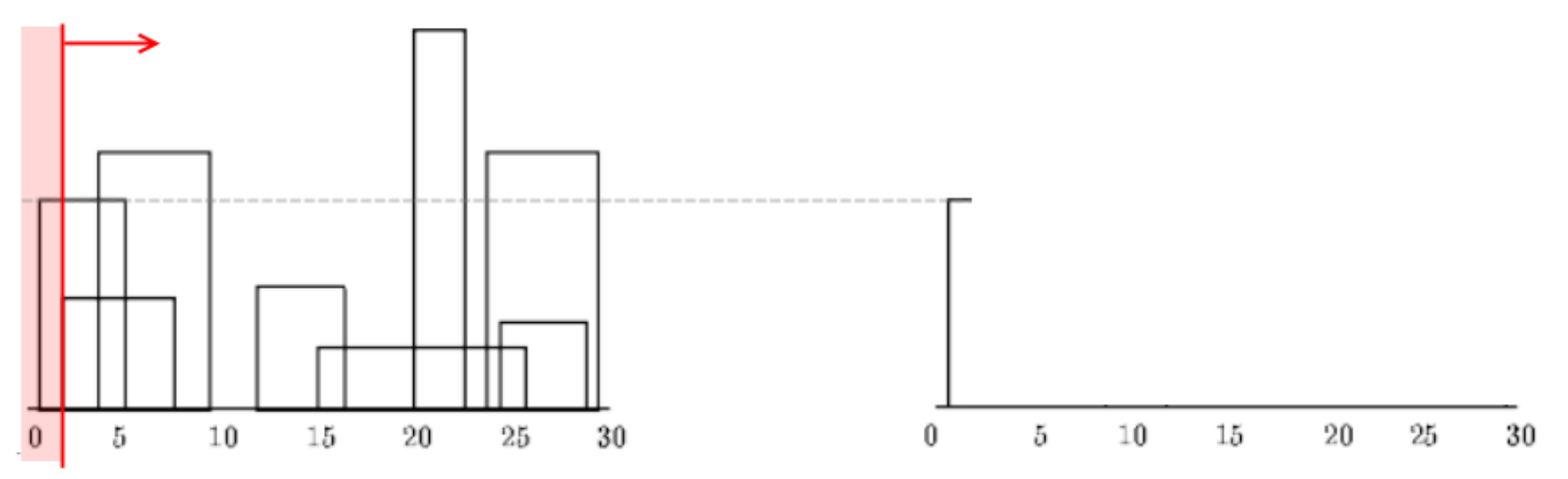
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



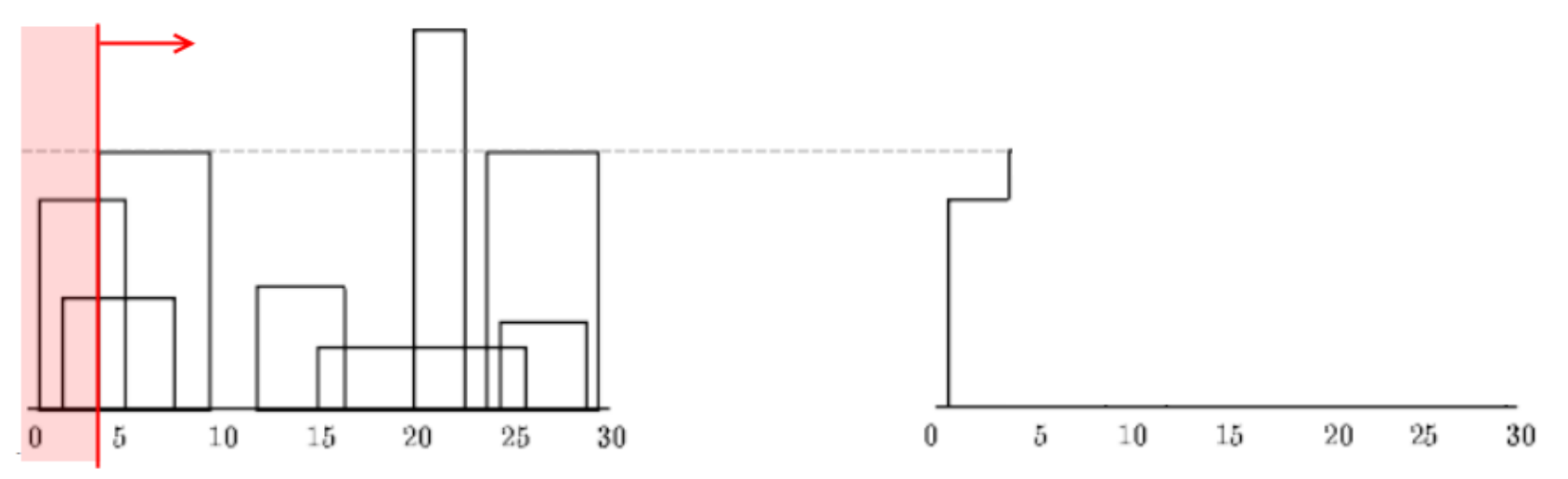
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



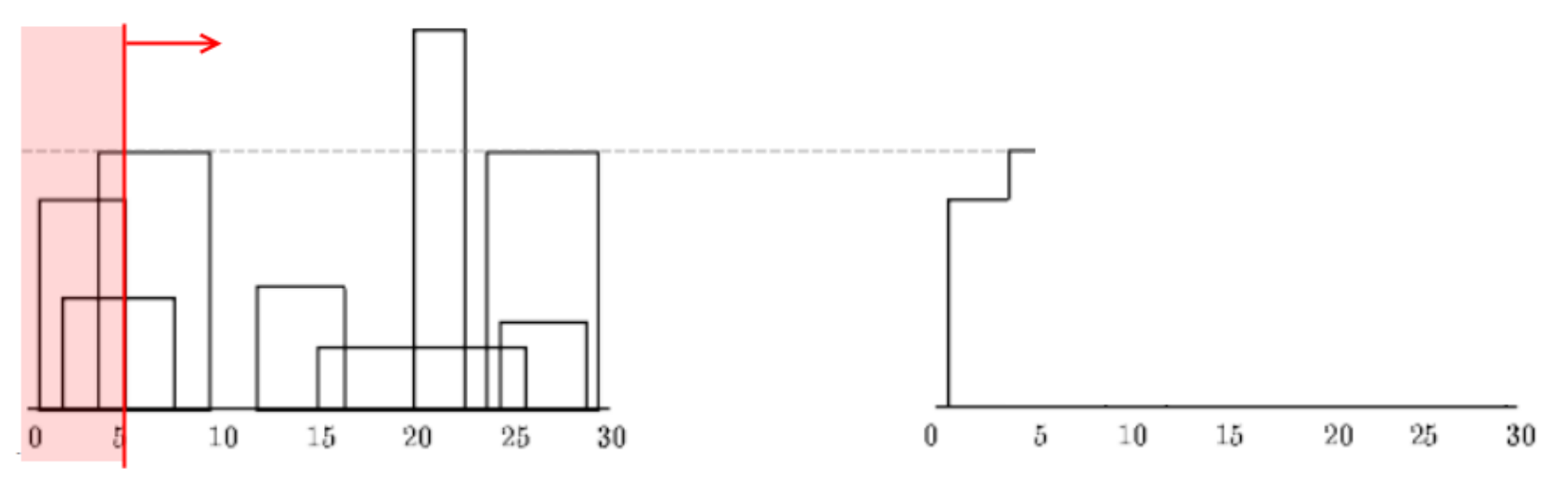
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



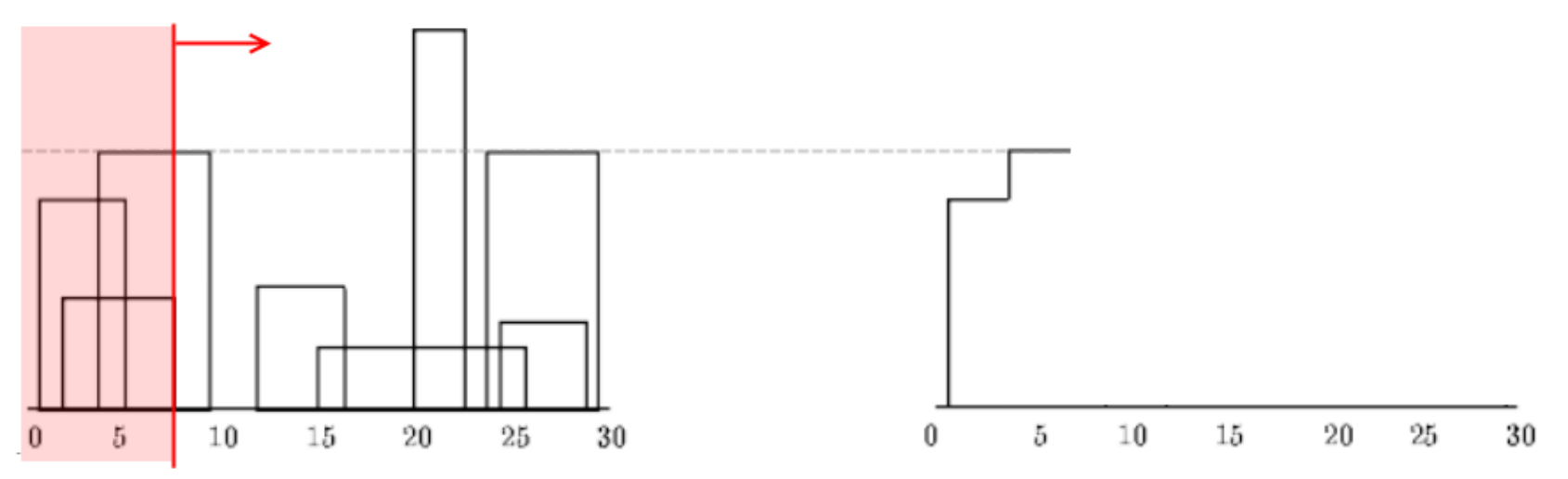
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



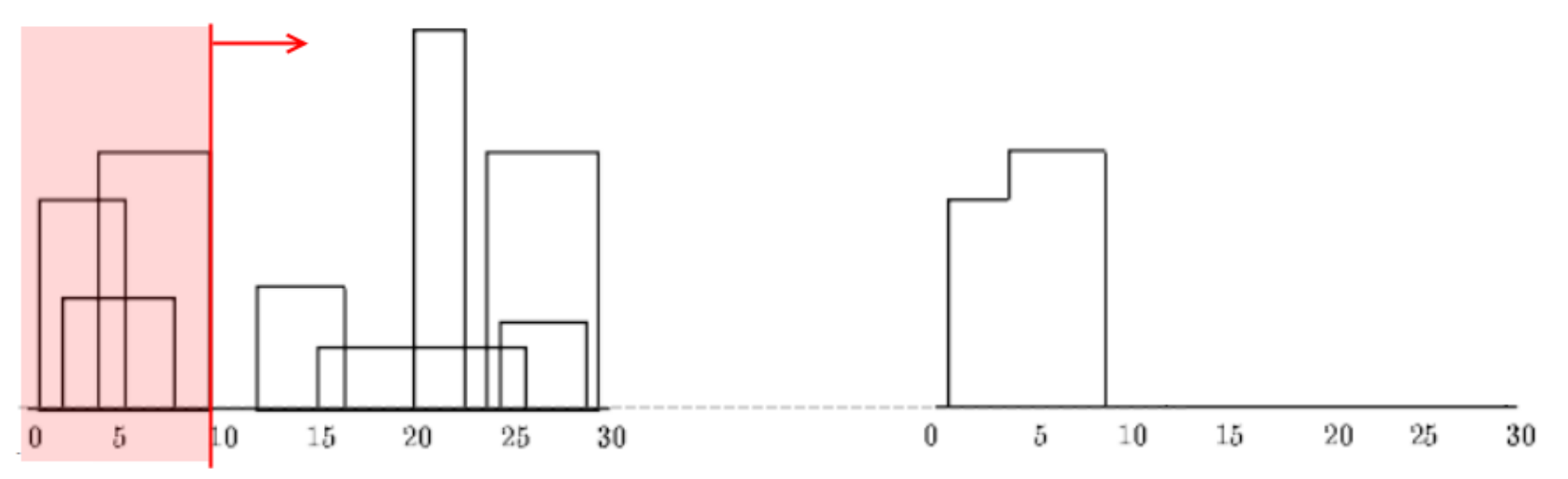
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



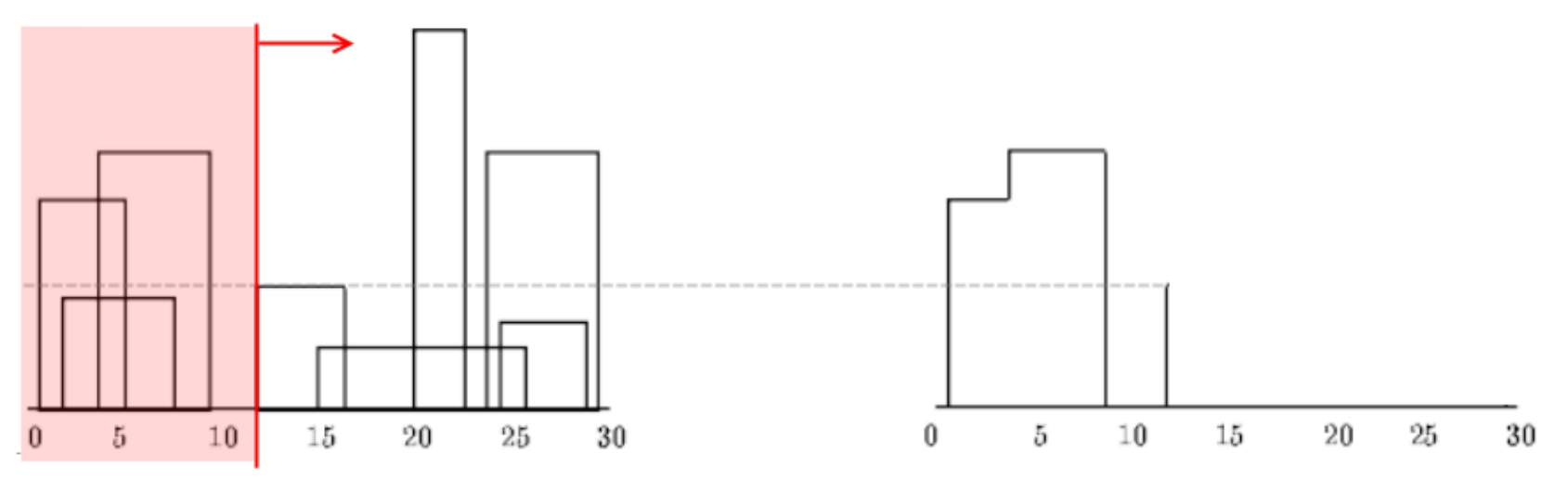
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



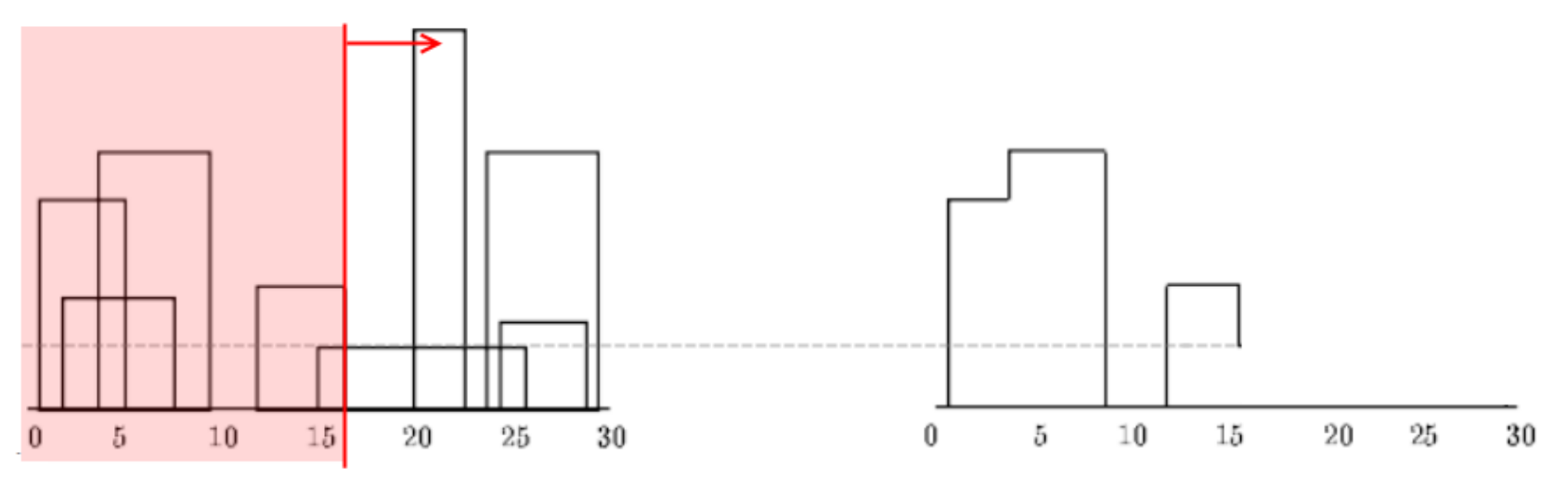
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



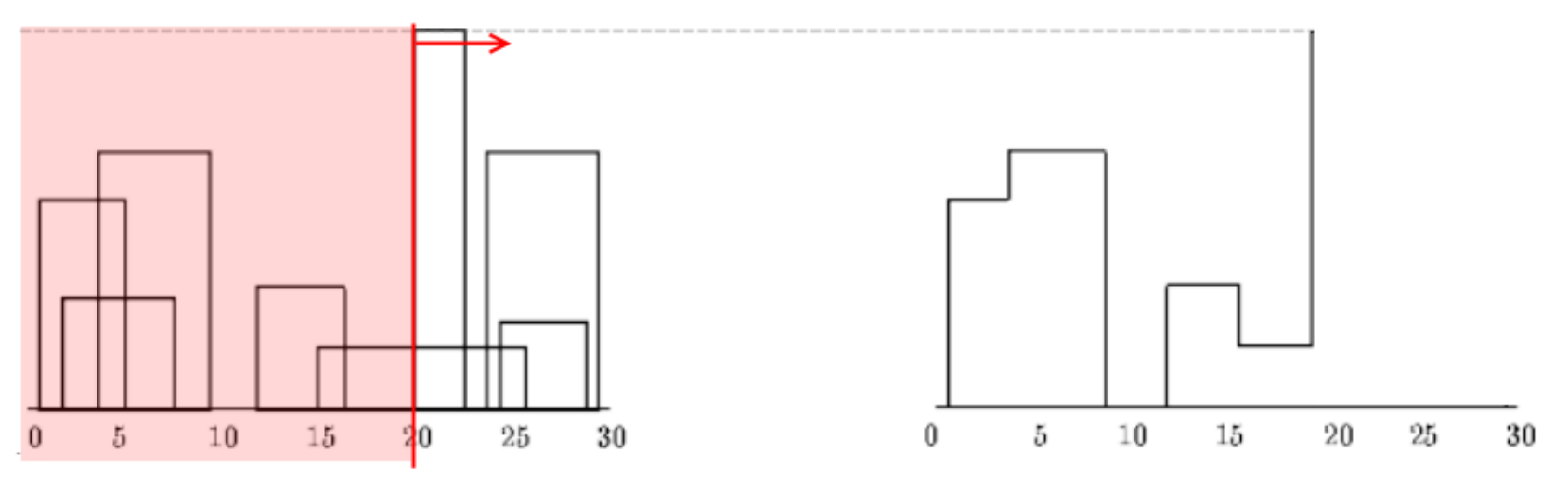
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



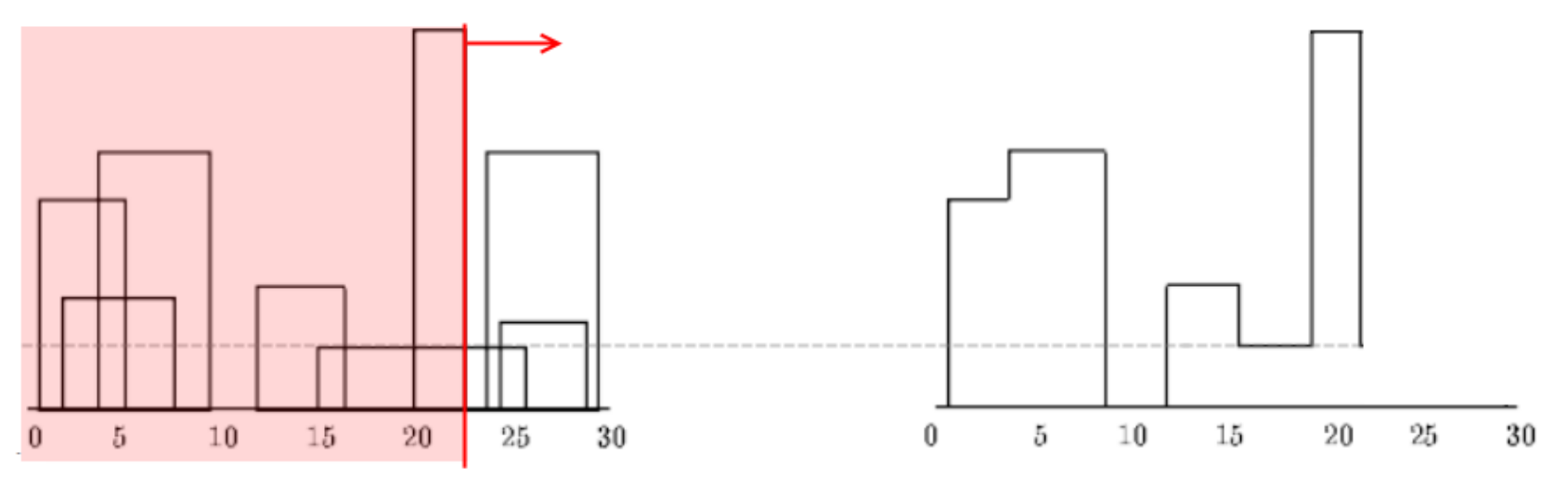
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



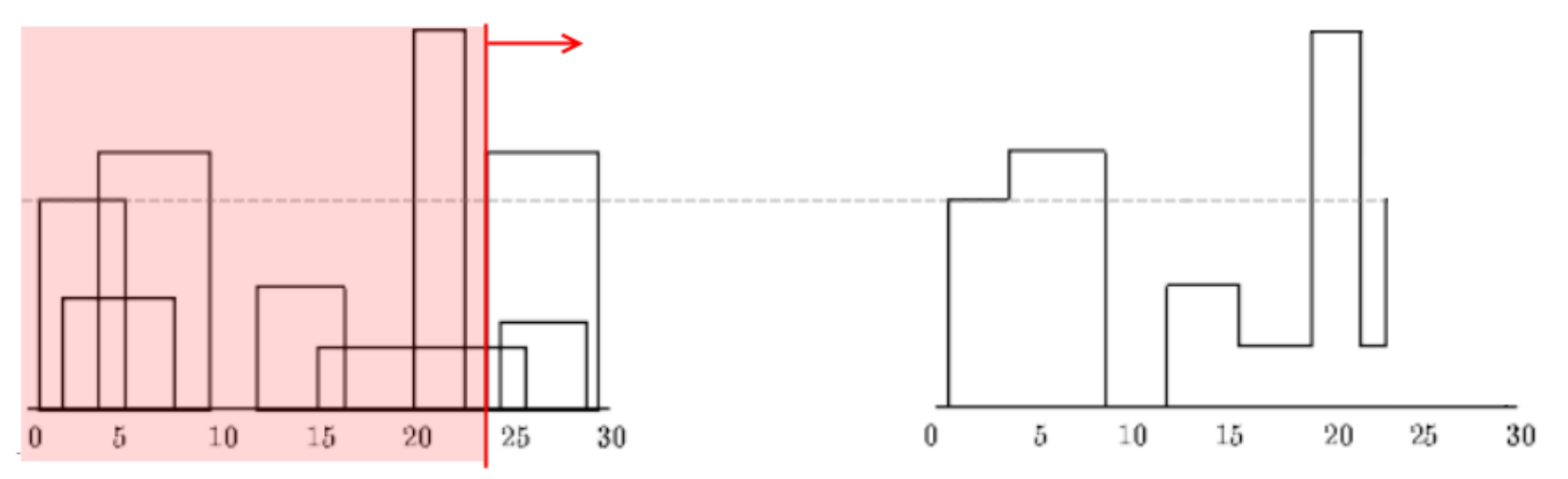
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



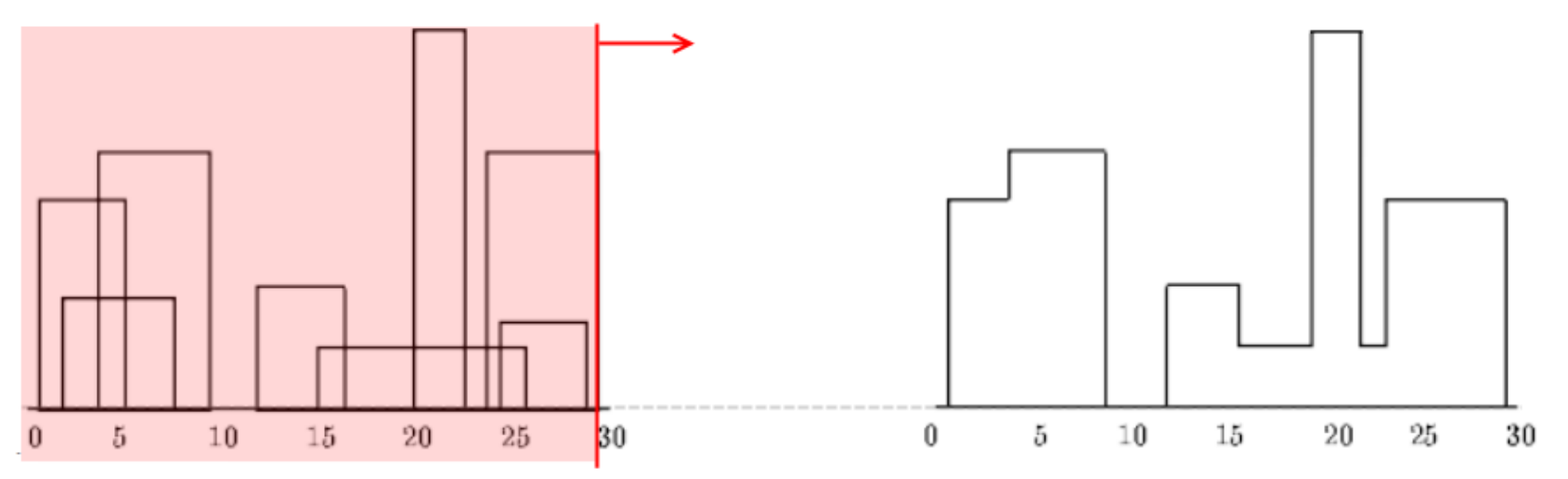
# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



# Problem 1: Solution

Imagine sweeping a vertical line from left to right.



# Problem 1: Solution

Imagine sweeping a vertical line from left to right.

State: multi-set of heights of rectangles on sweep line

Events: left and right bounds of a rectangle

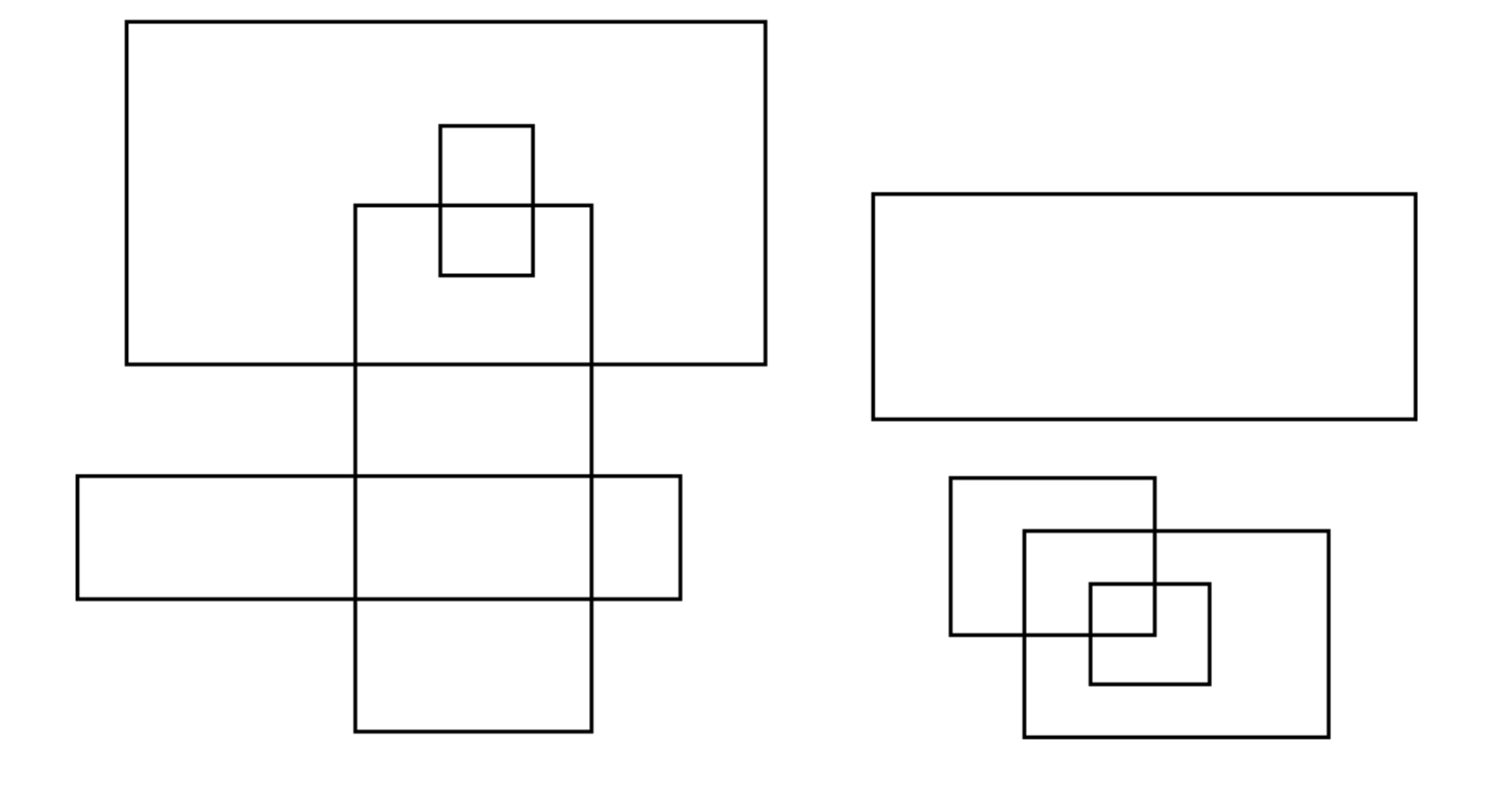
Actions:

- Left bound of rectangle $\rightarrow$ insert height into the set
- Right bound of rectangle $\rightarrow$ erase height from set
- Whenever height changes, add new segment(s) to skyline
- Process all events at same $x$ before modifying skyline!

Time complexity: $O(n \log n)$ for sorting $n$ events, $O(\log n)$ per event, so $O(n \log n)$ in total

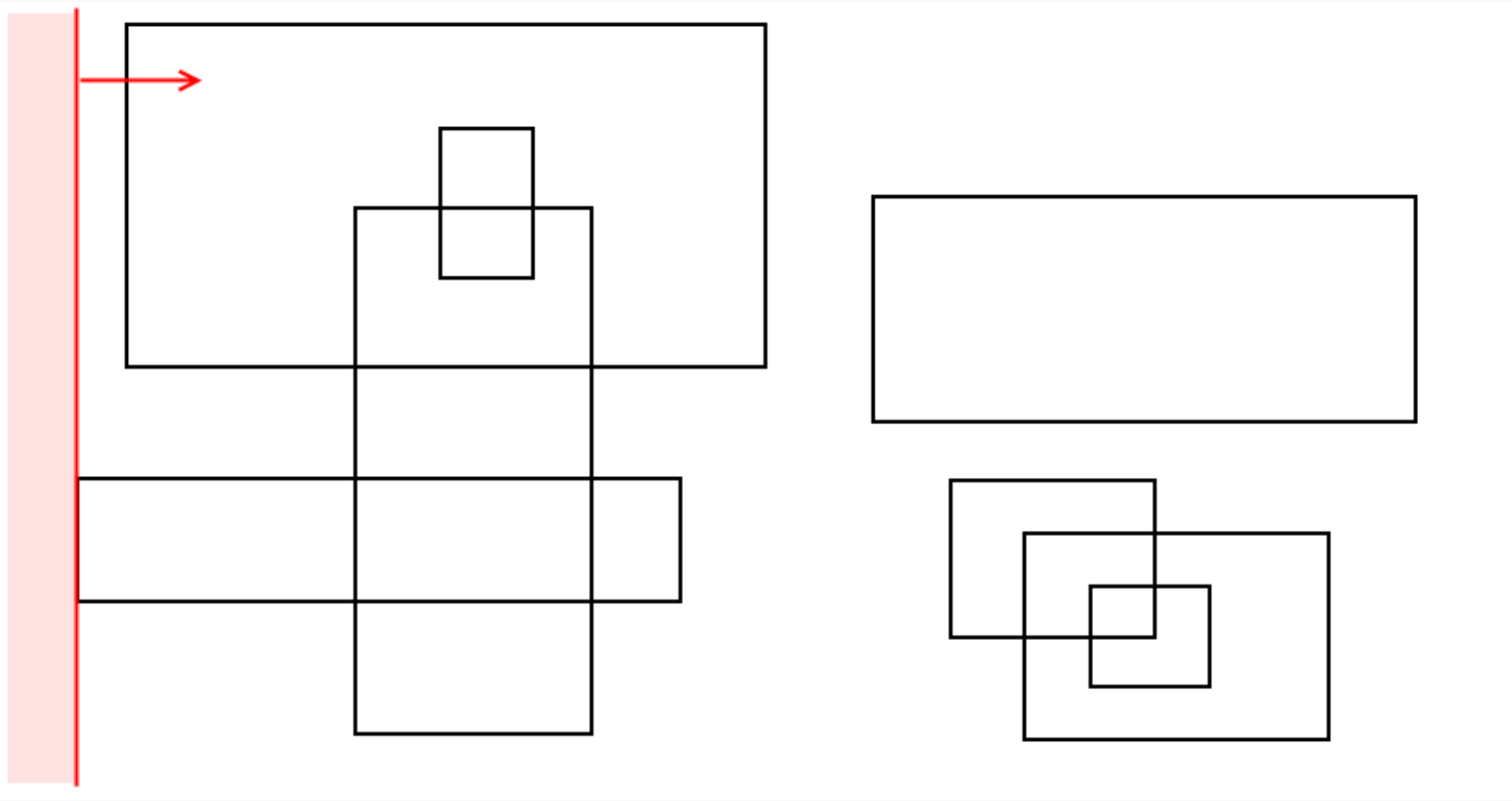
# Problem 2: Rectangle Area Union

Compute how much area these rectangles cover in total!



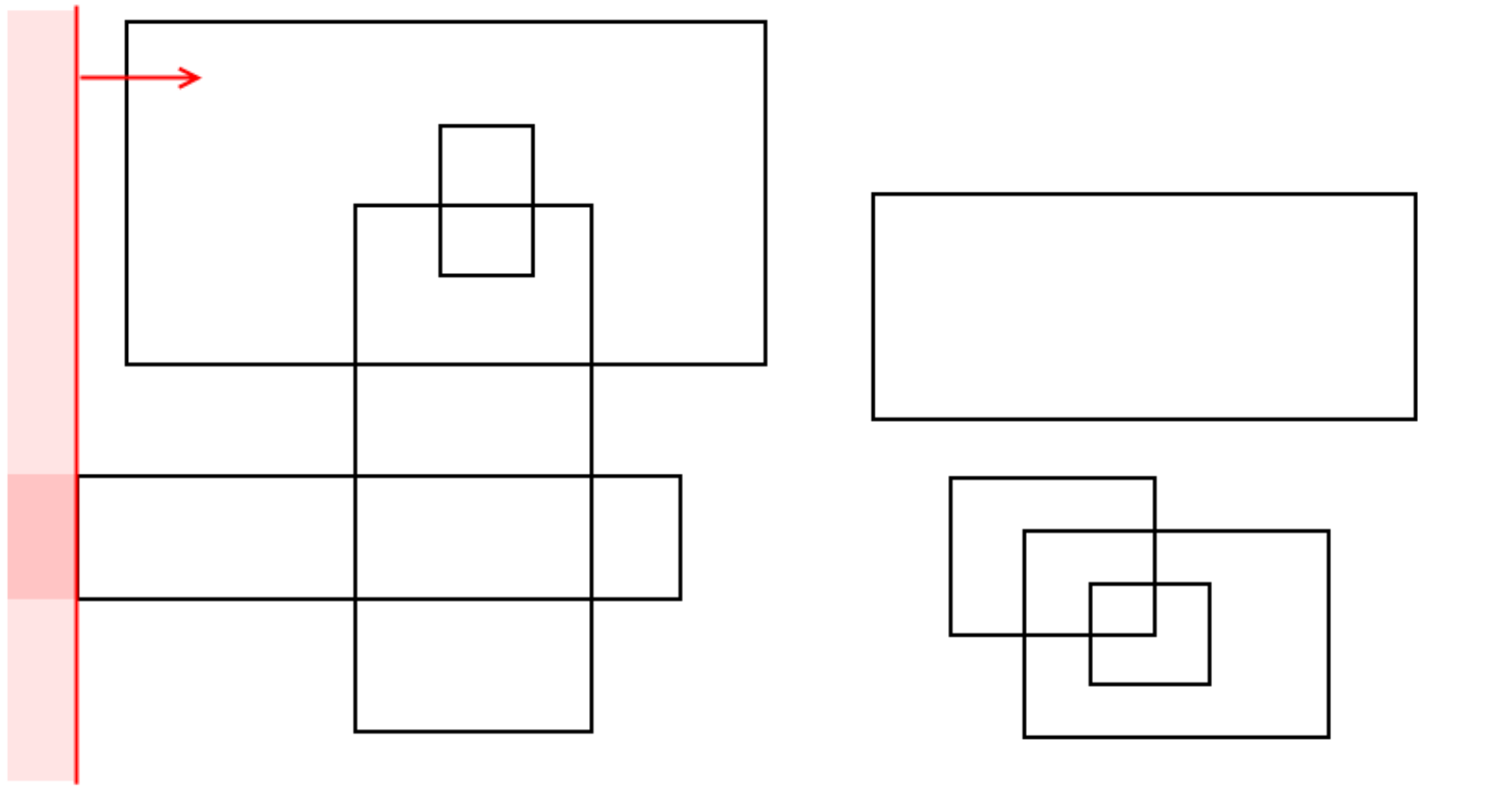
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



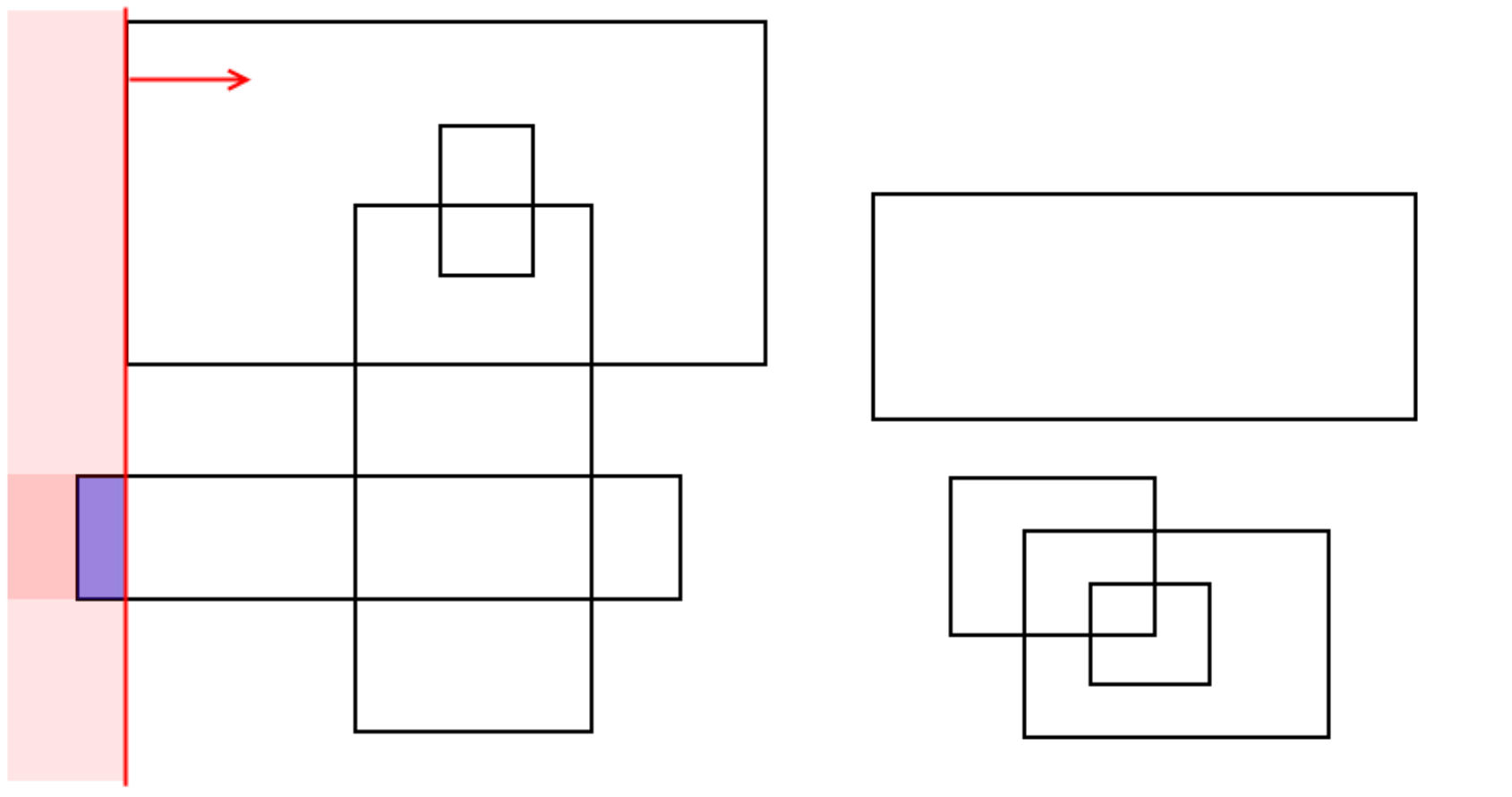
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



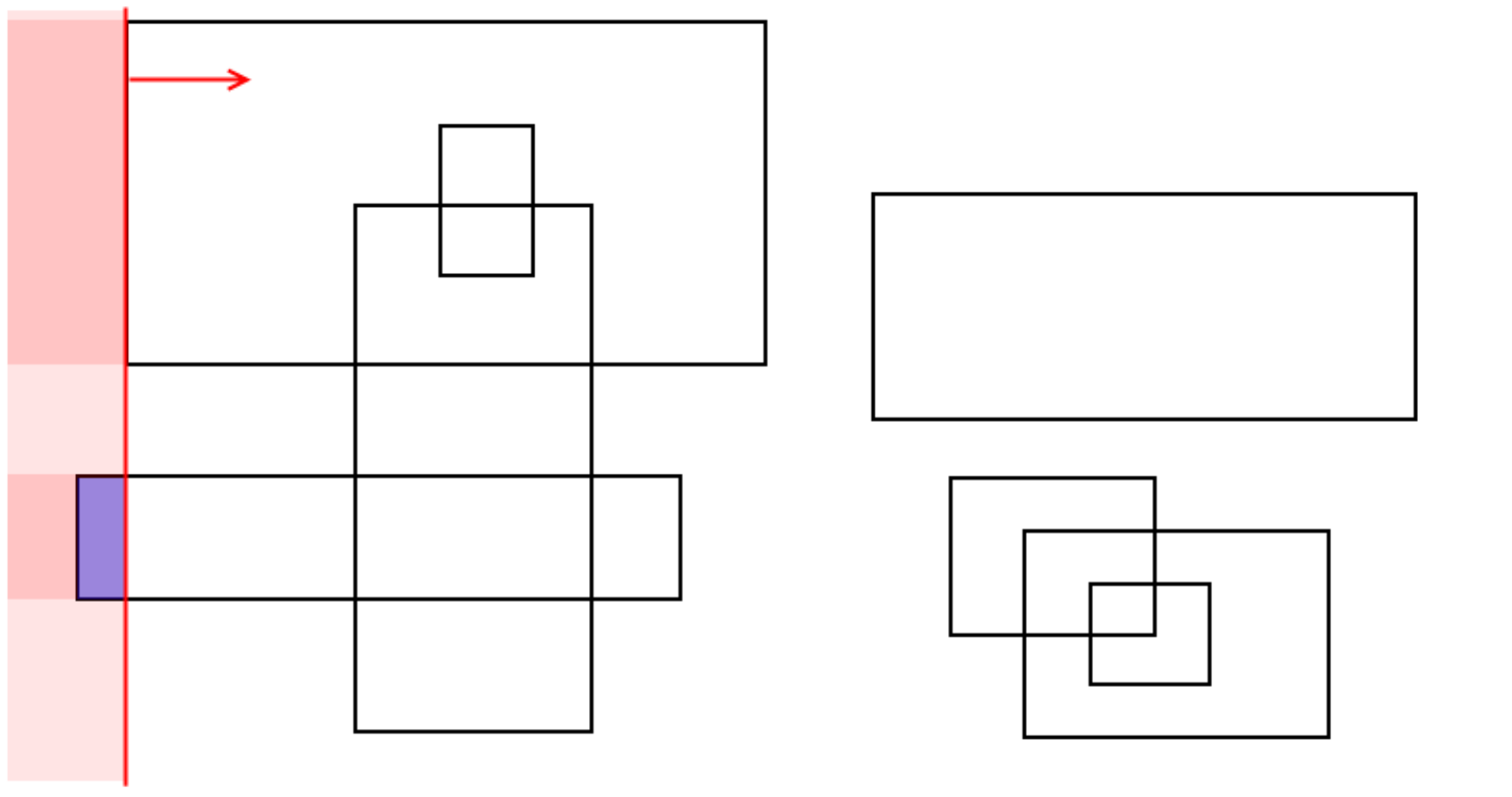
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



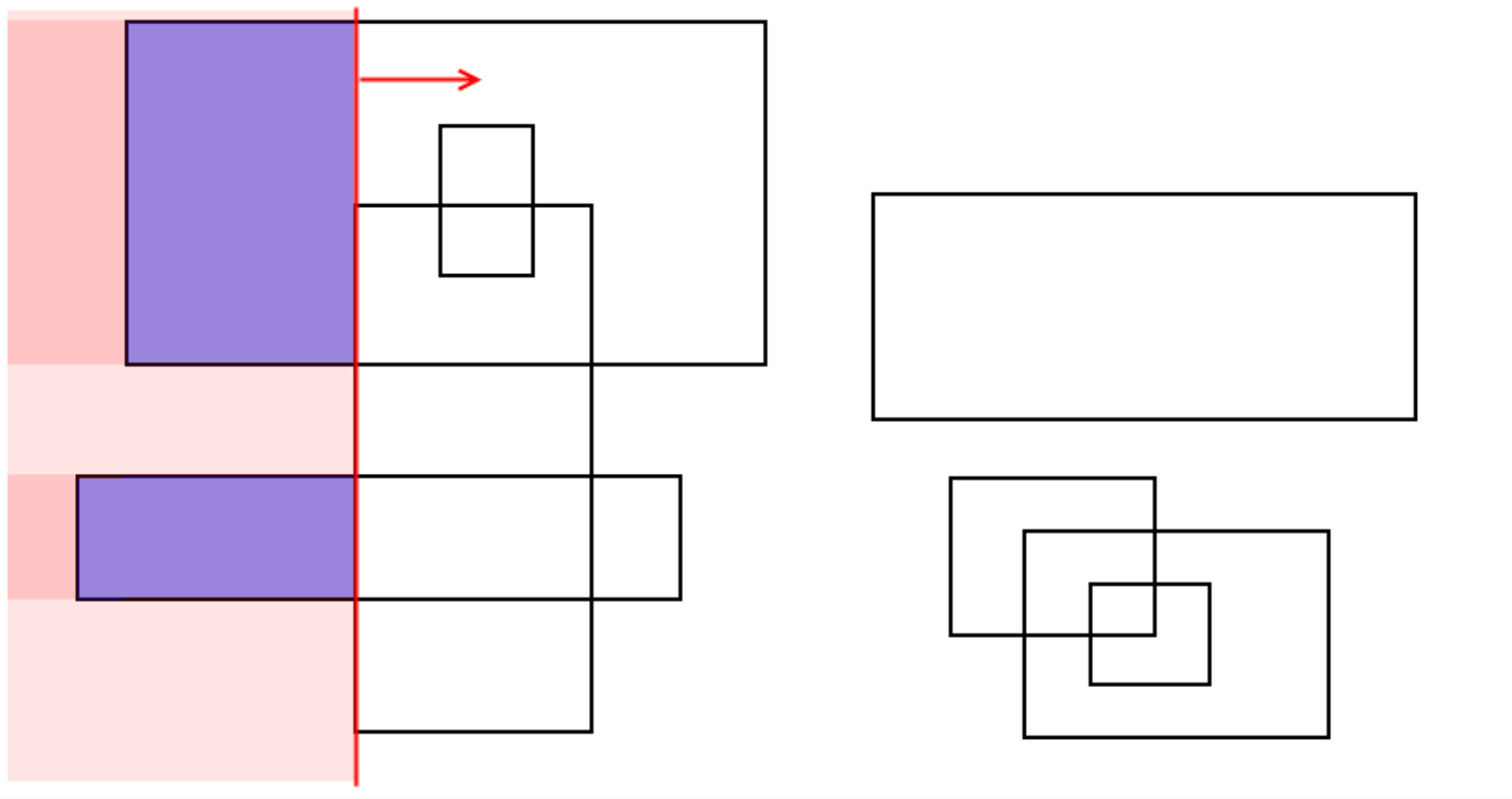
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



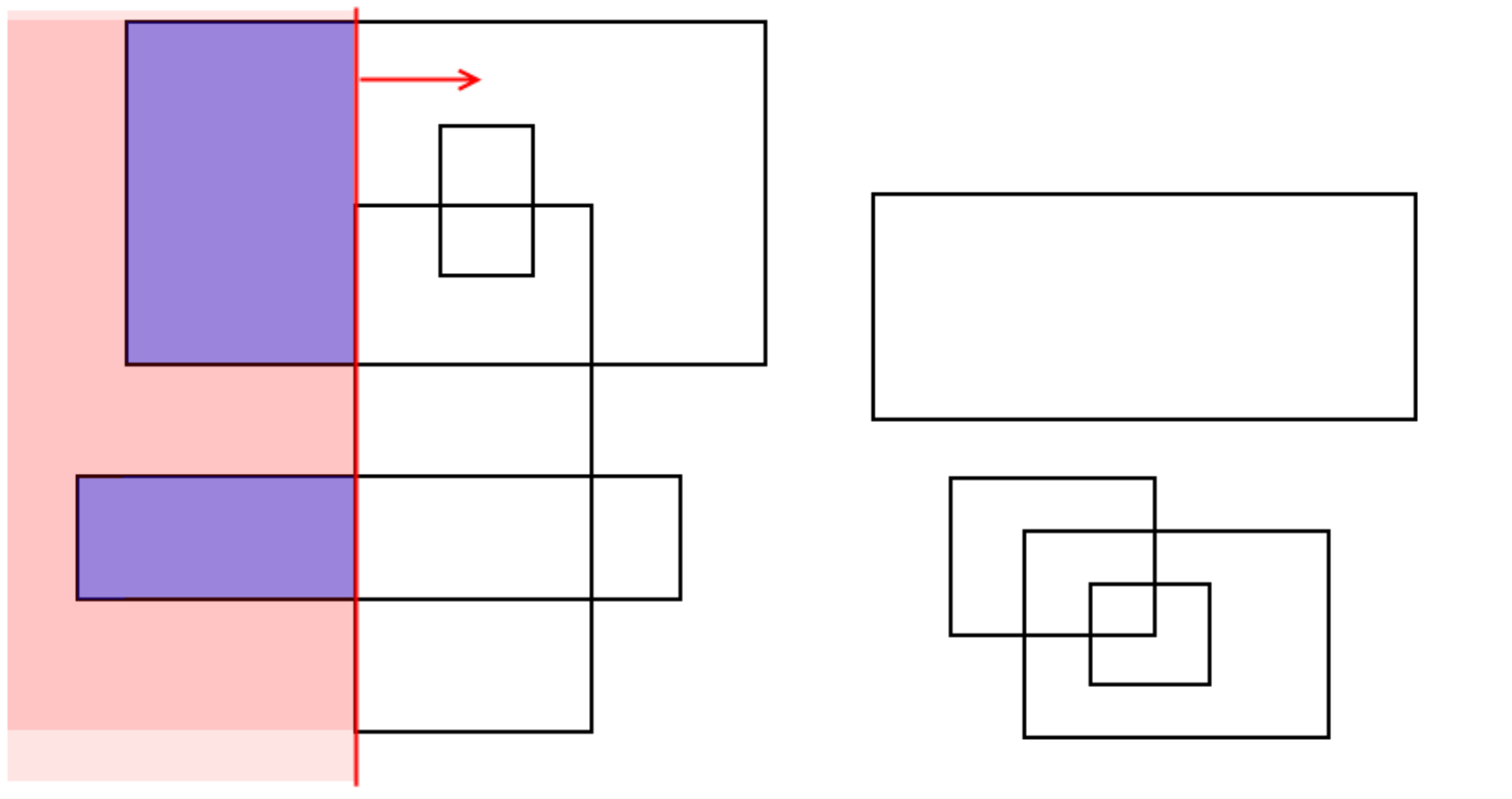
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



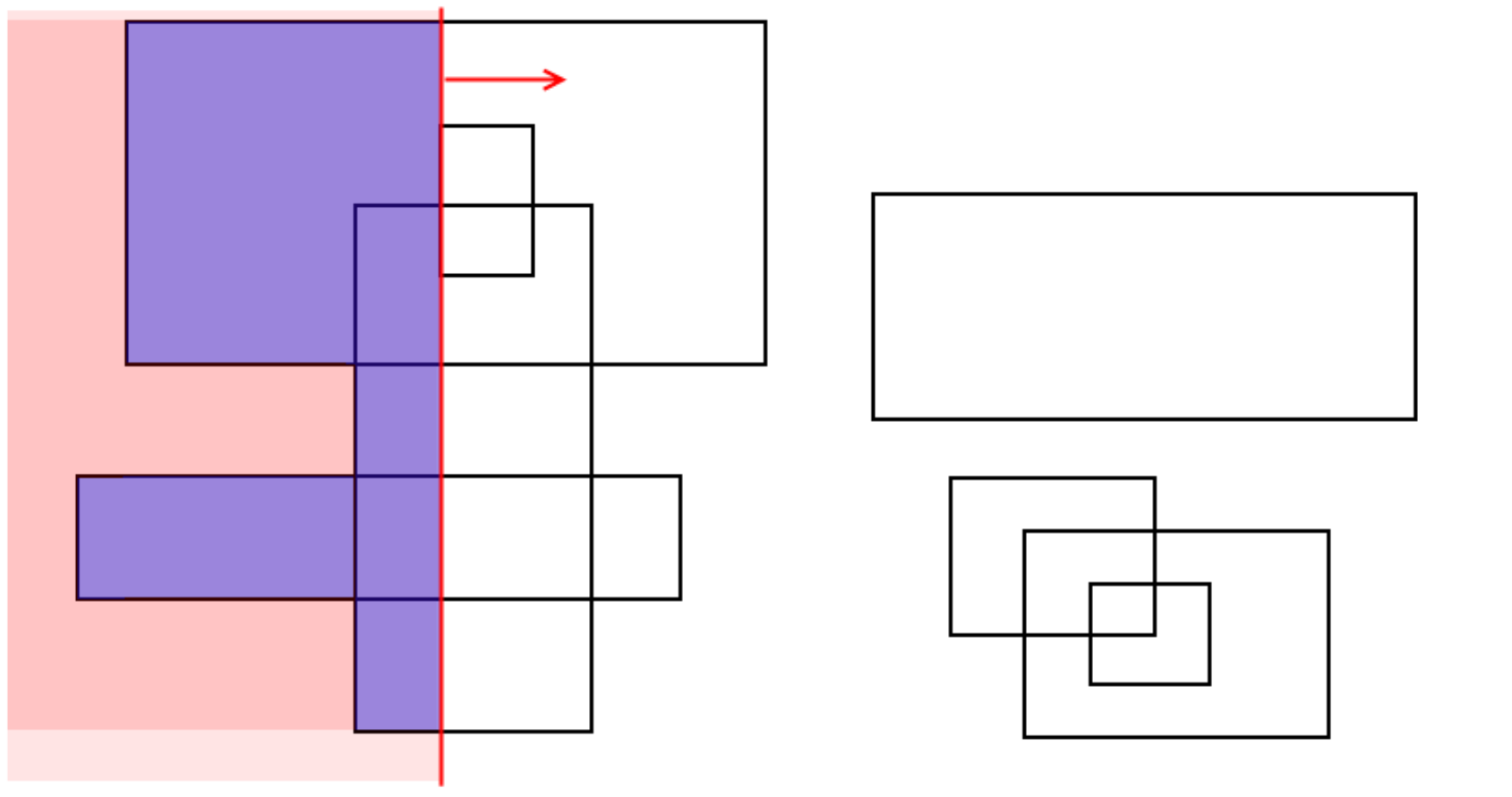
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



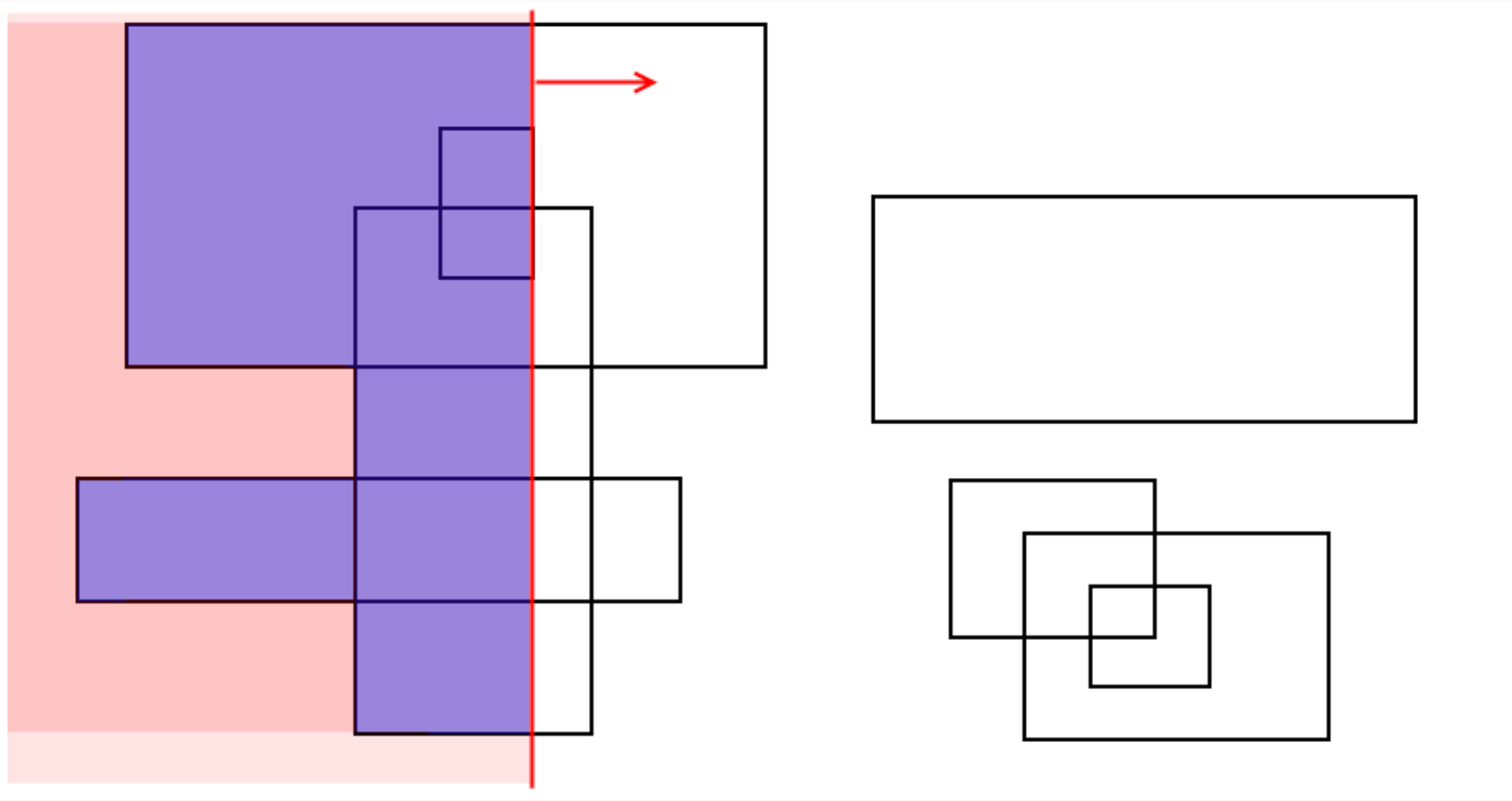
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



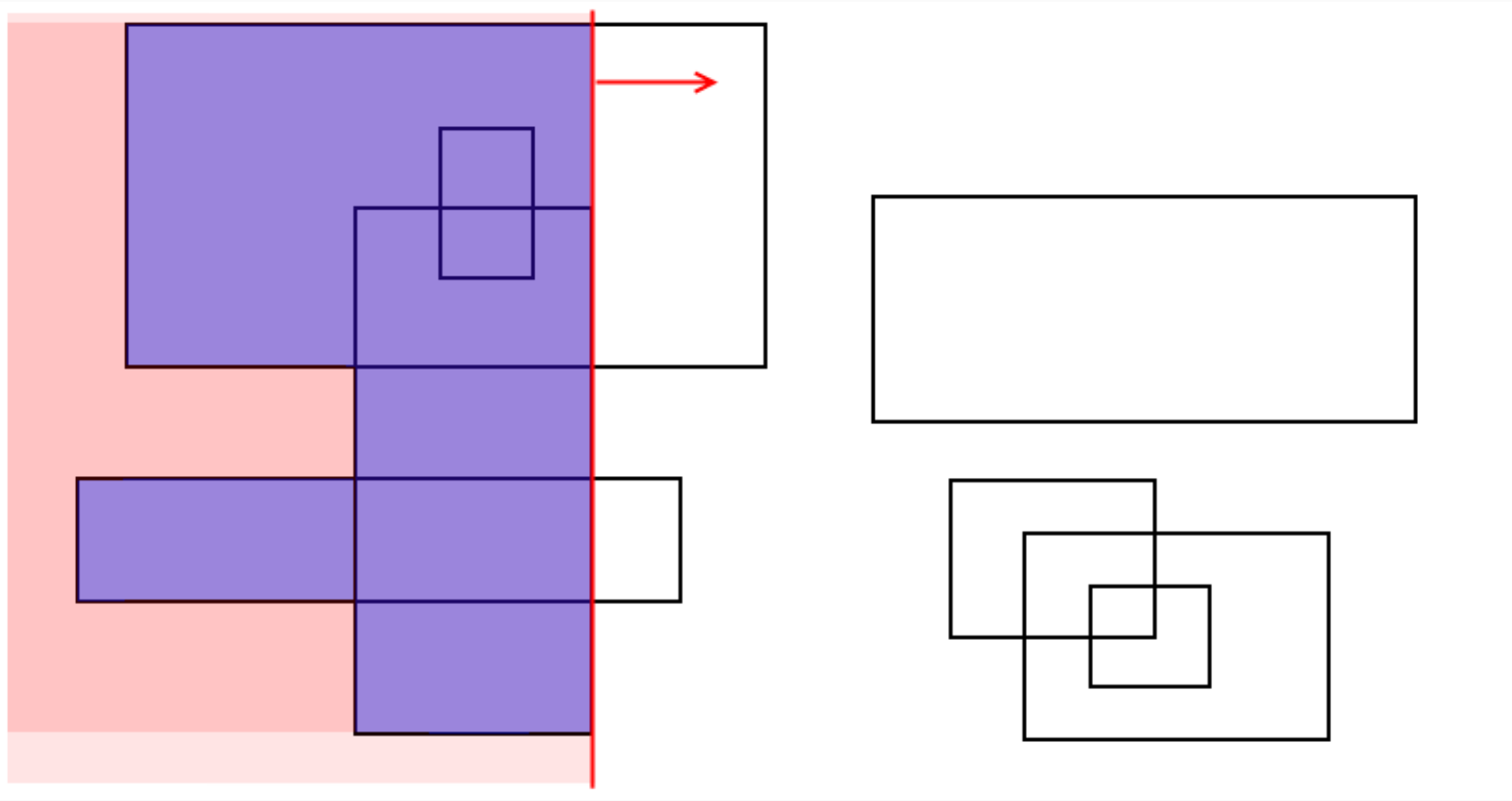
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



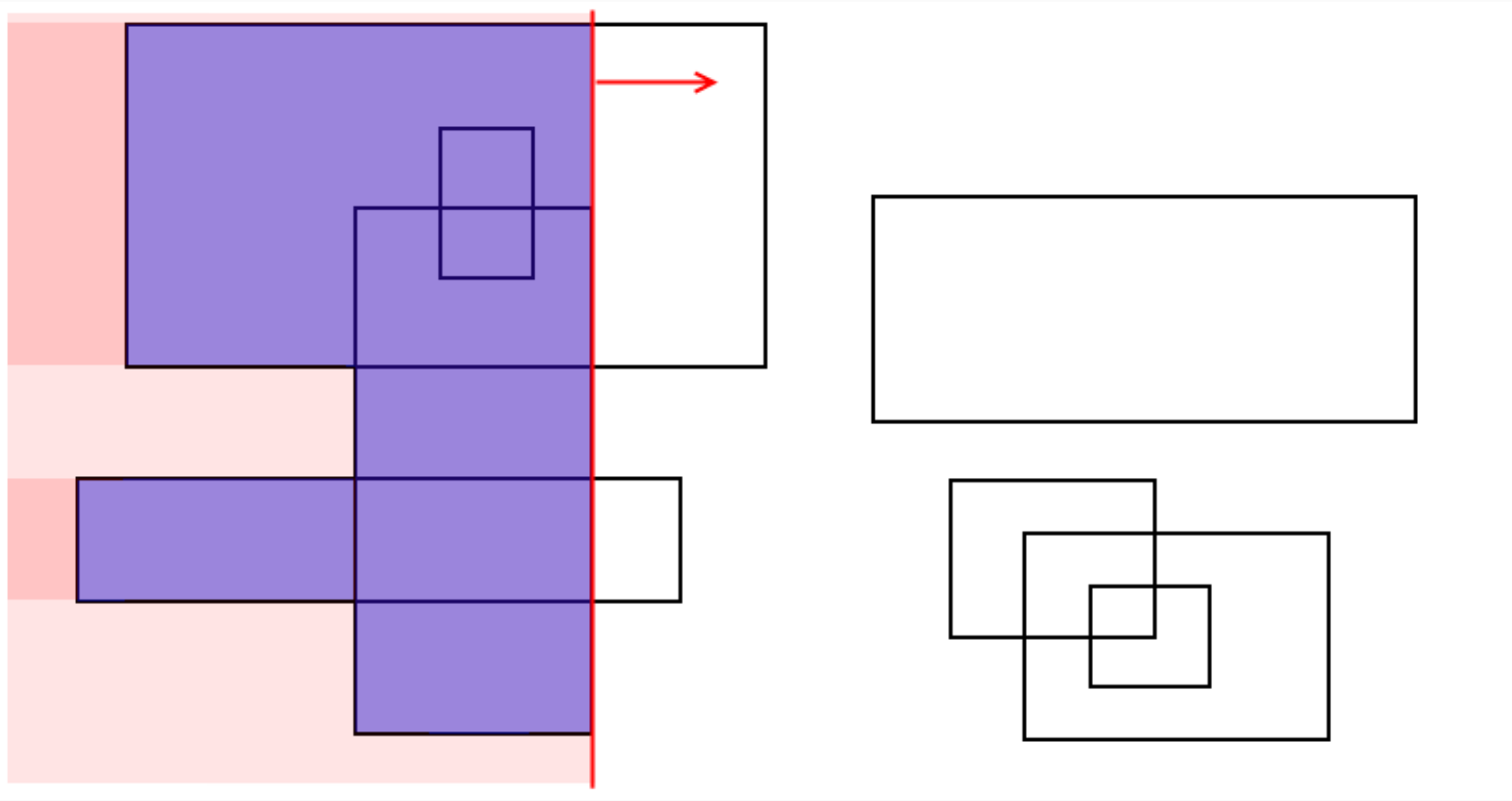
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



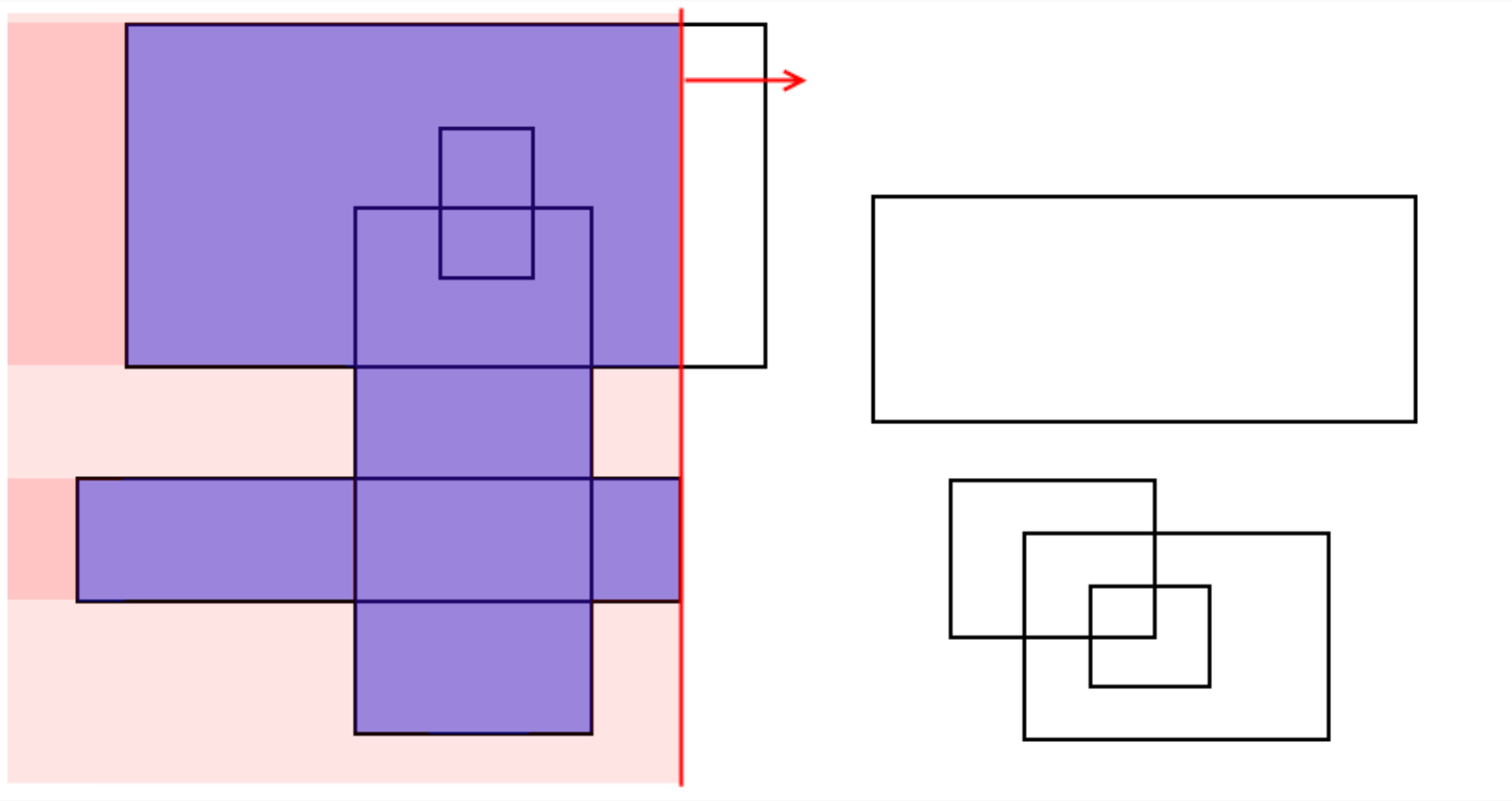
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



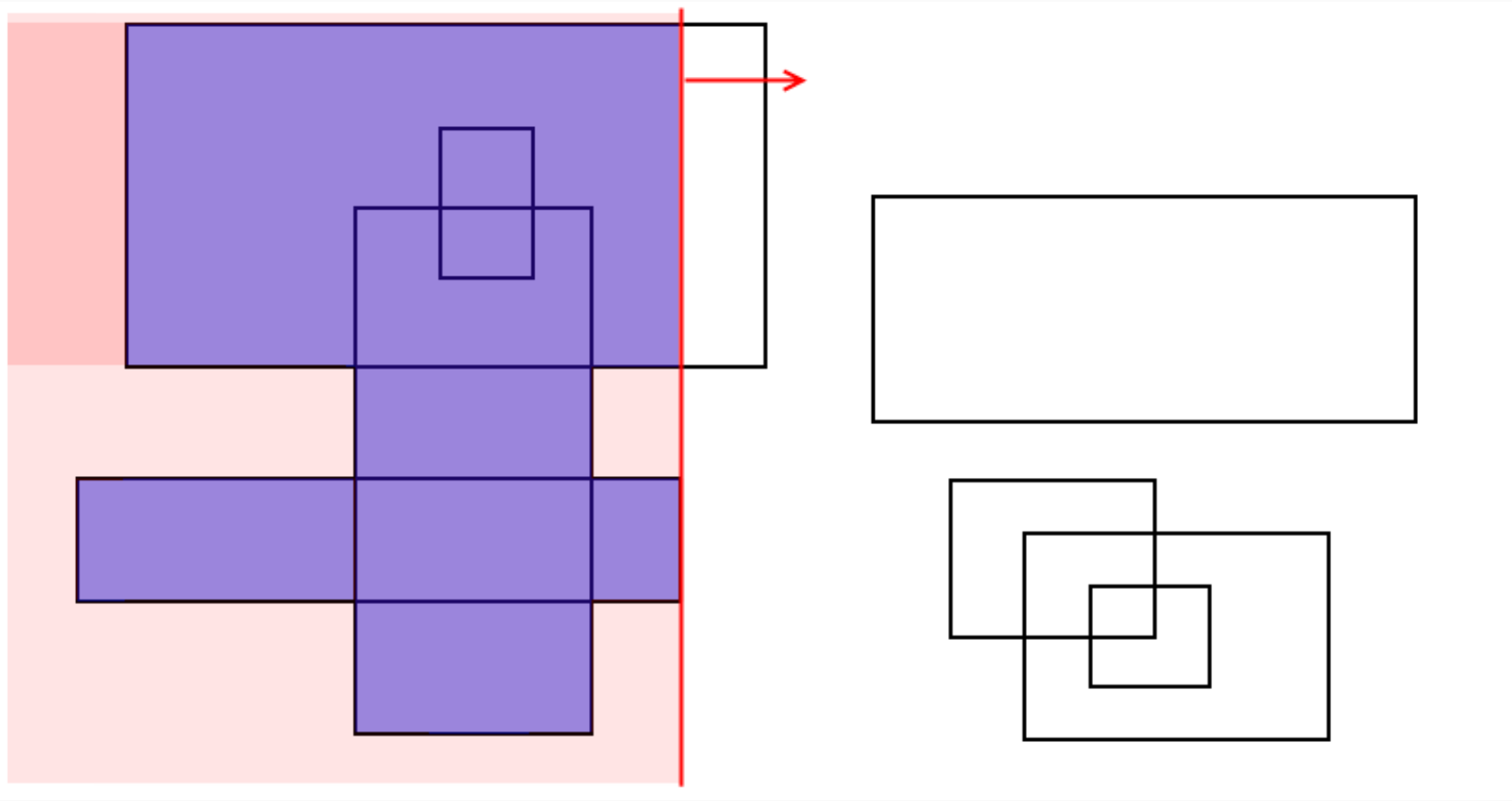
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



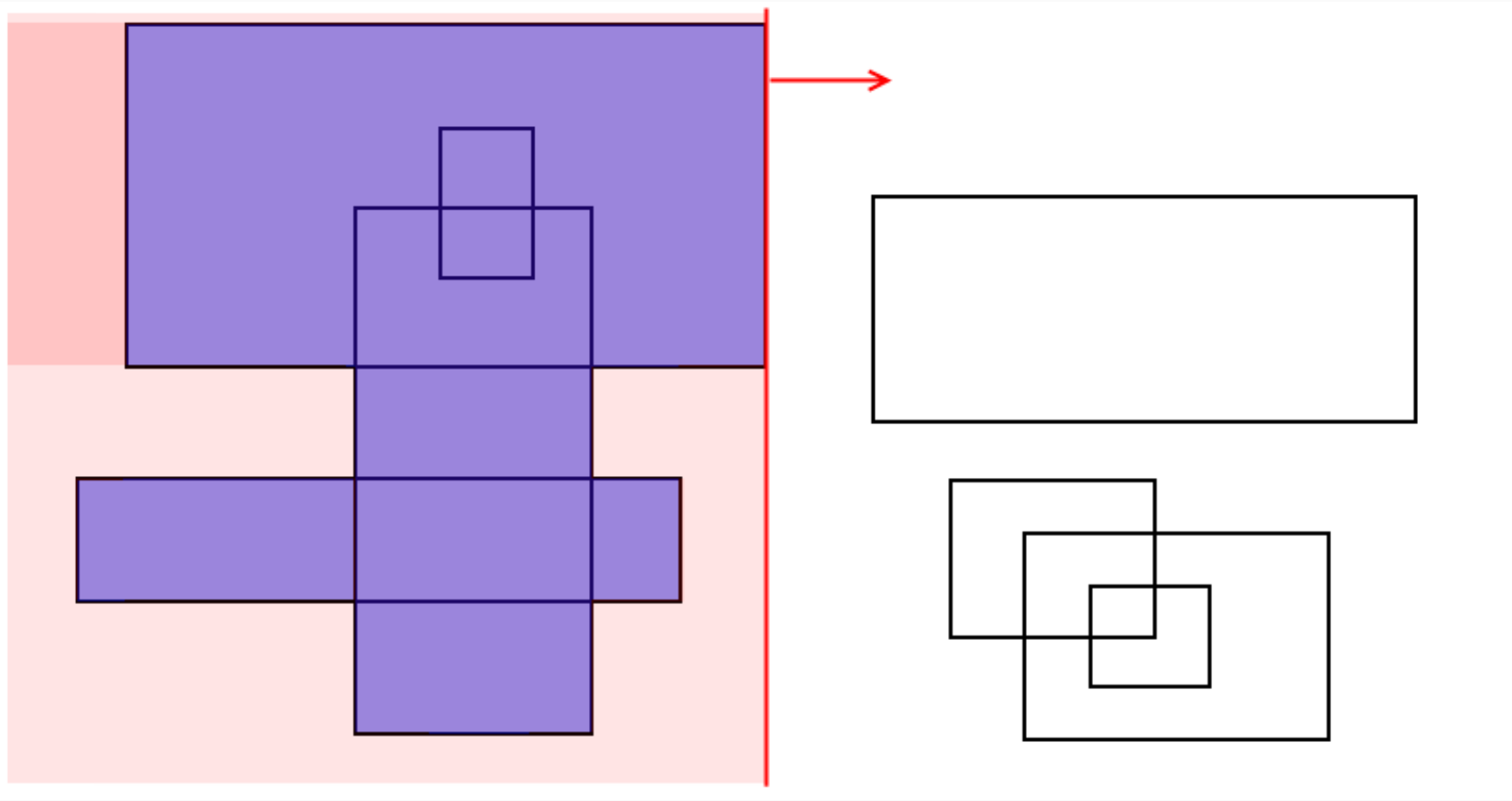
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



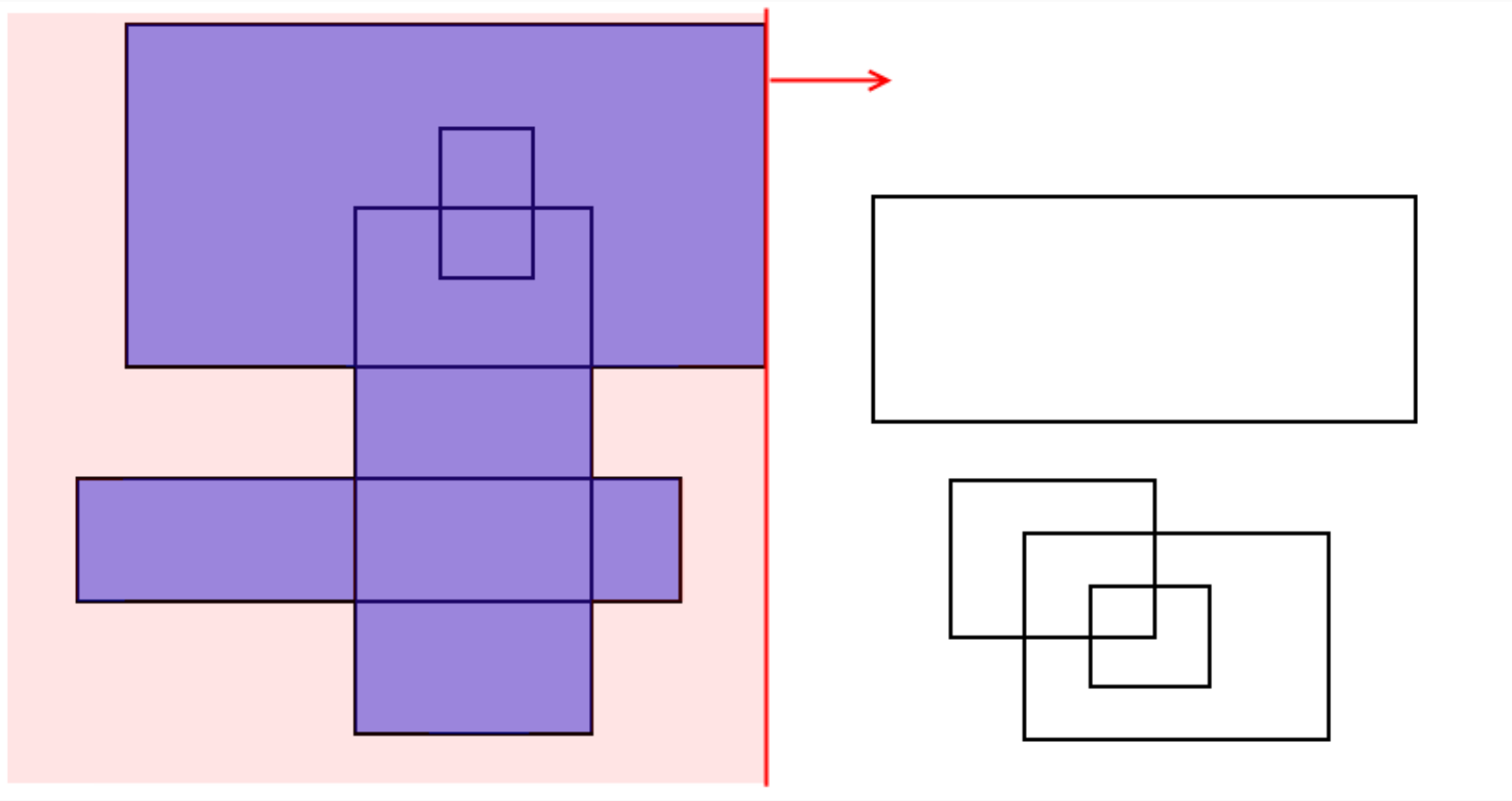
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



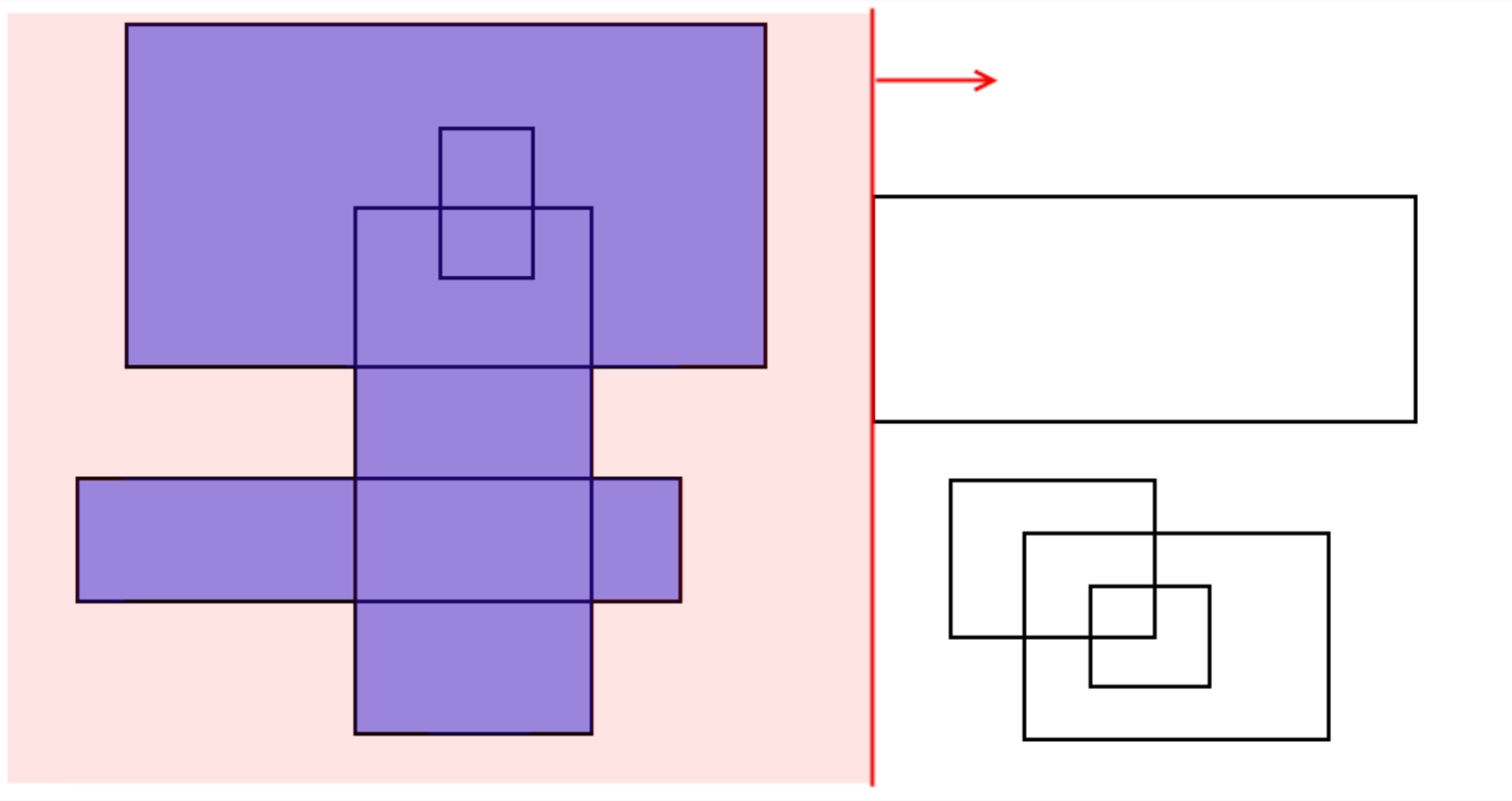
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



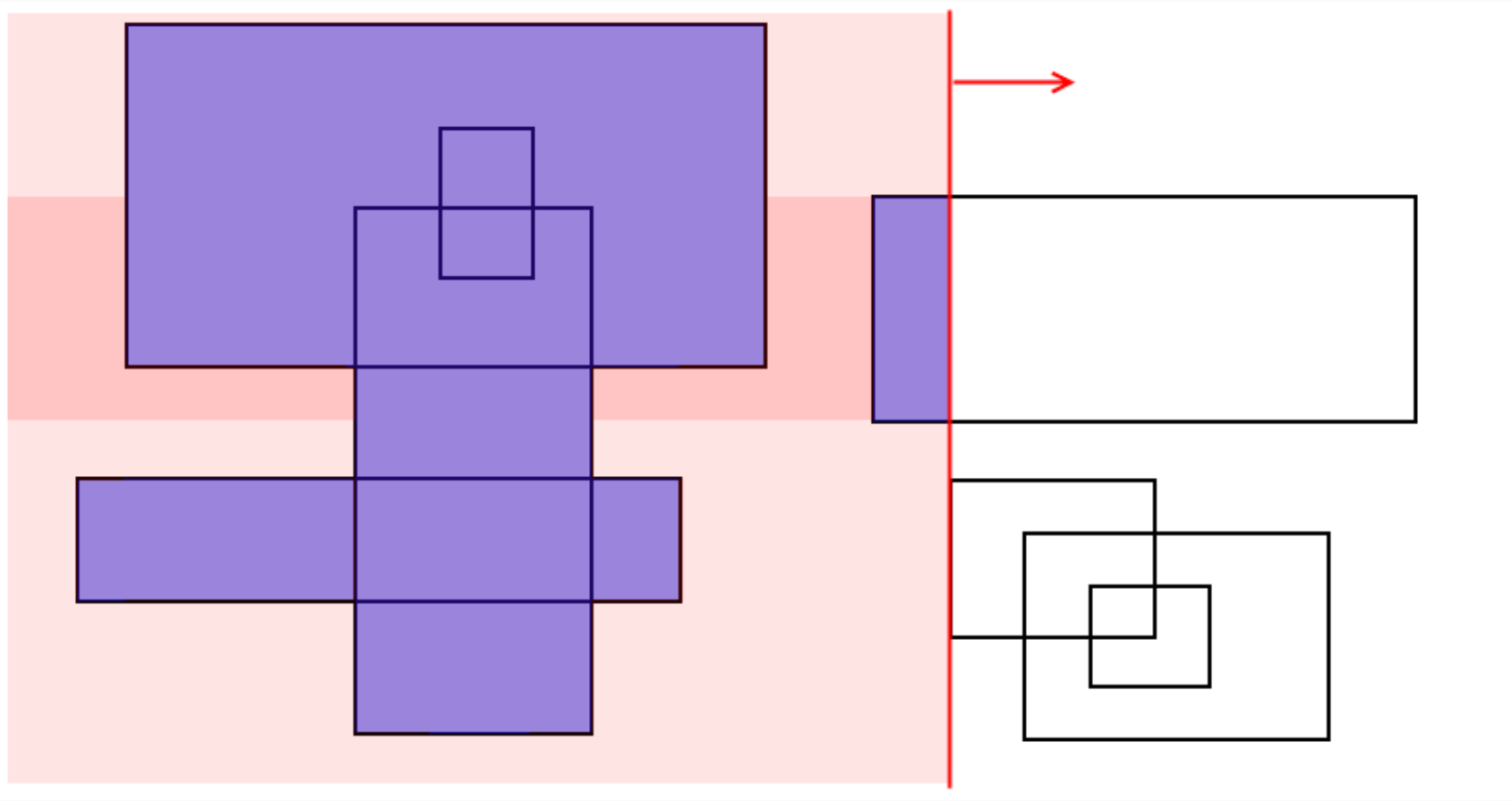
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



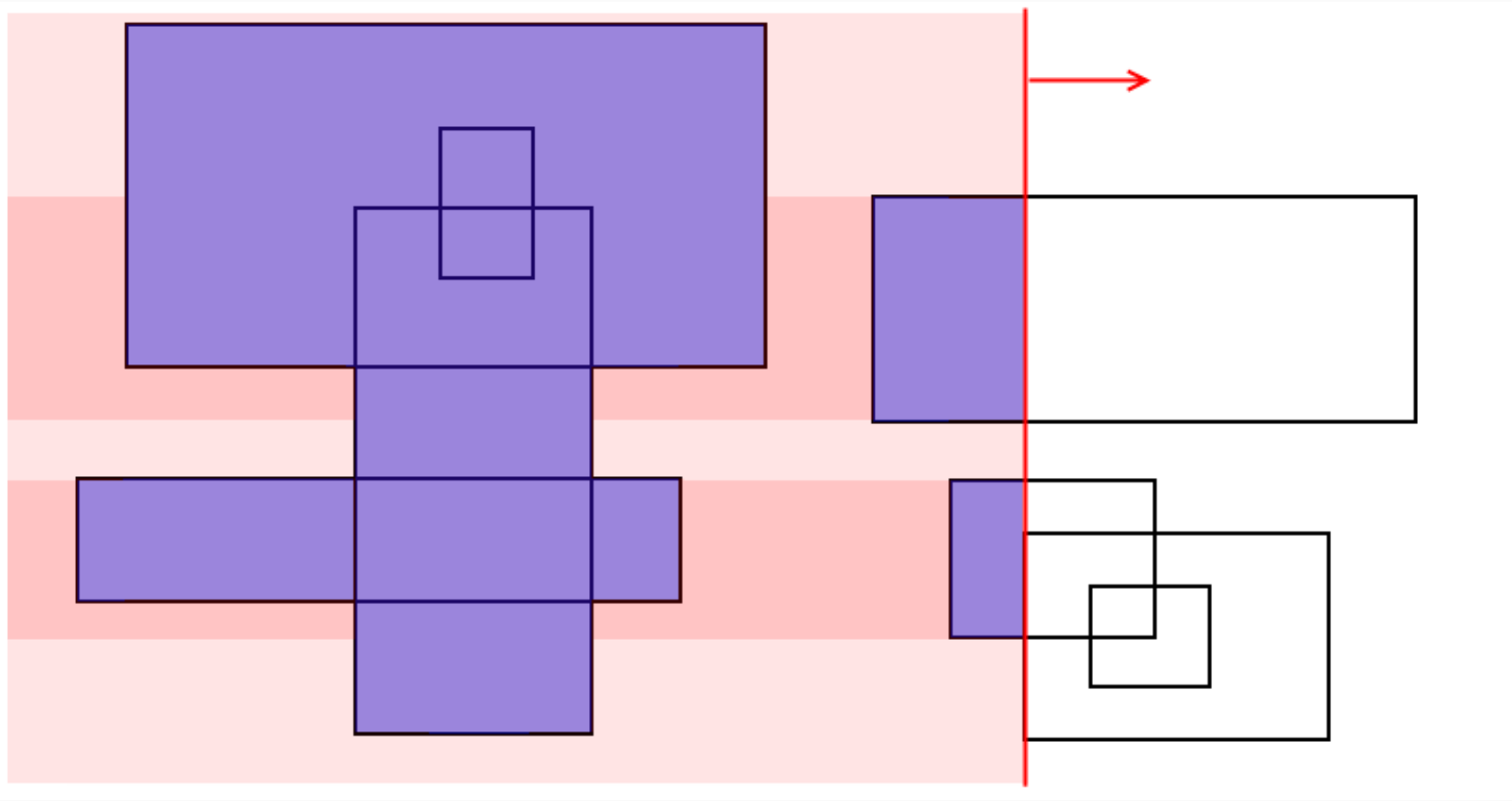
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



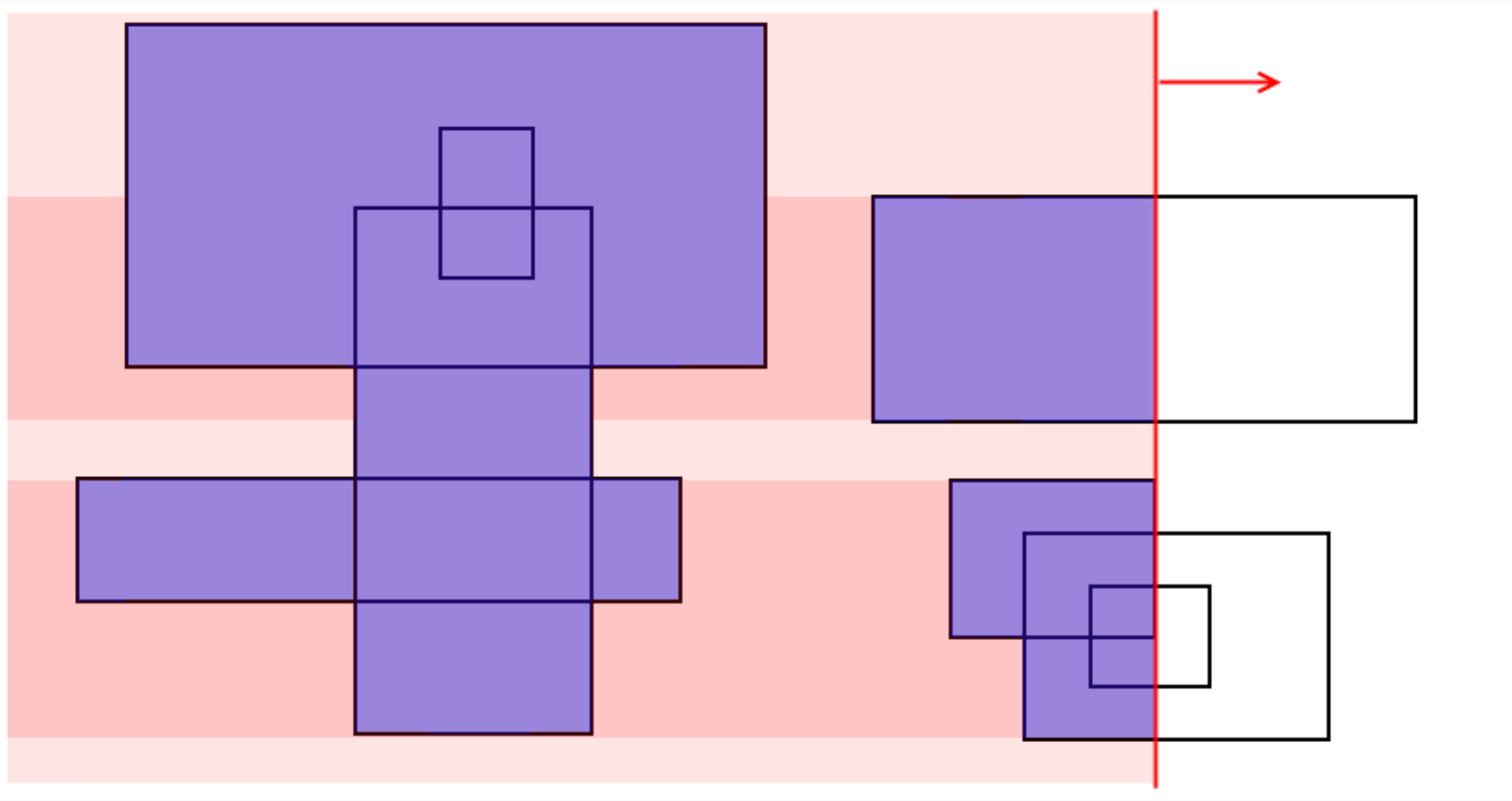
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



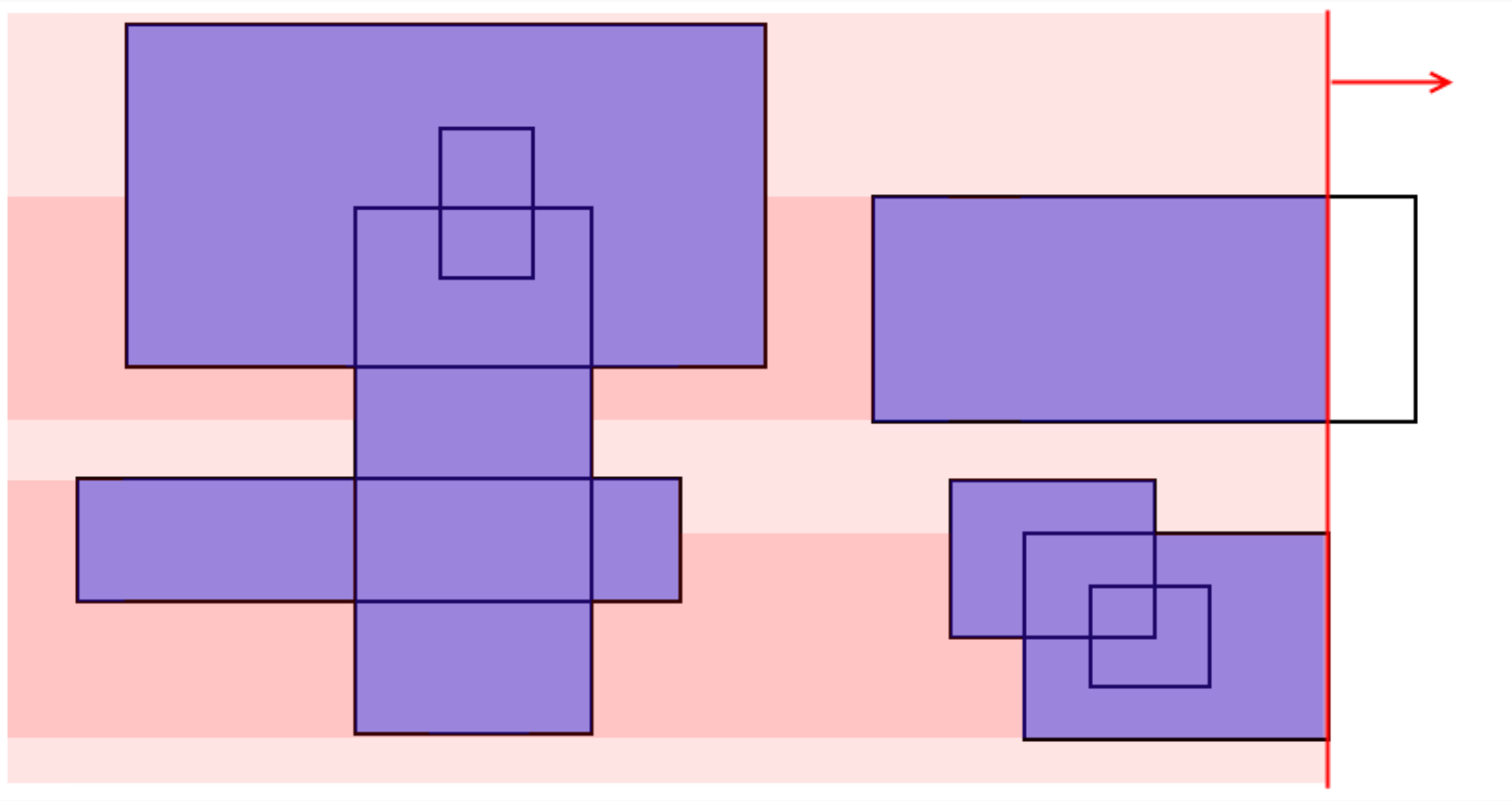
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



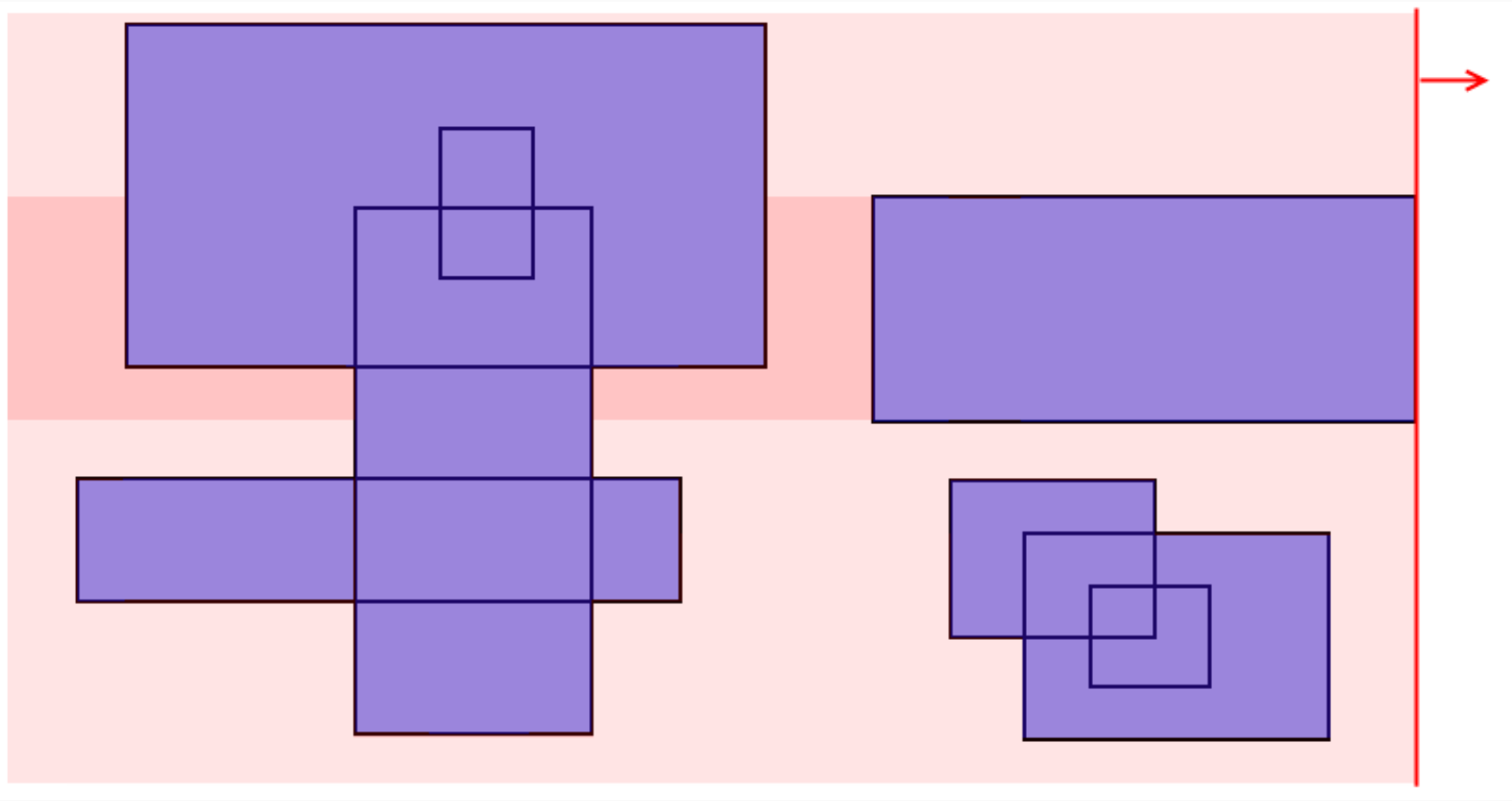
# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



# Problem 2: Solution

Imagine sweeping a vertical line from left to right.



# Problem 2: Solution

Imagine sweeping a vertical line from left to right.

State: intervals of sweep line covered by rectangle

Events: left and right bounds of a rectangle

Actions:

- Left bound of rectangle $\rightarrow$ insert segment
- Right bound of rectangle $\rightarrow$ erase segment
- Before events at $x$, add $\Delta x \cdot$(segment length union) to area

# Problem 2: Solution

Imagine sweeping a vertical line from left to right.

State: intervals of sweep line covered by rectangle

Events: left and right bounds of a rectangle

Actions:

- Left bound of rectangle $\rightarrow$ insert segment
- Right bound of rectangle $\rightarrow$ erase segment
- Before events at $x$, add $\Delta x \cdot$(segment length union) to area

Problem: need a data structure to efficiently do segment insertion, deletion and segment length union

# Problem 2: Solution

Imagine sweeping a vertical line from left to right.

State: intervals of sweep line covered by rectangle

Events: left and right bounds of a rectangle

Actions:

- Left bound of rectangle $\rightarrow$ insert segment
- Right bound of rectangle $\rightarrow$ erase segment
- Before events at $x$, add $\Delta x \cdot$(segment length union) to area

Problem: need a data structure to efficiently do segment insertion, deletion and segment length union

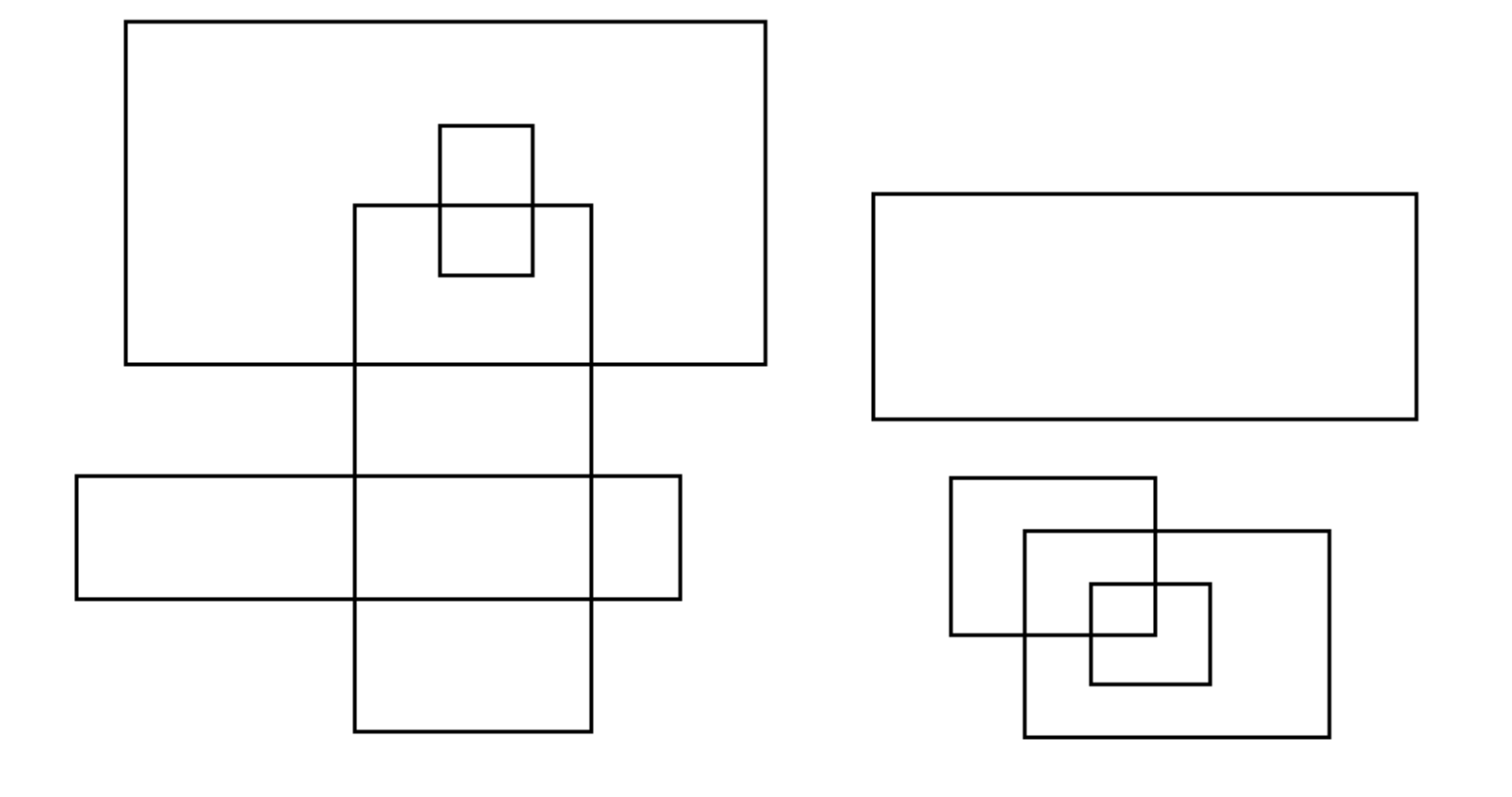
Solution: segment tree! Each node keeps a count of how many segments inserted here, and total coverage of its range.

Time complexity: $O(n \log n)$

# Problem 3: Rectangle Perimeter Union

Compute length of perimeter of union of these rectangles!

Hint: sweep twice



# Problem 3: Solution

Sweep left to right to get length of horizontal segments.

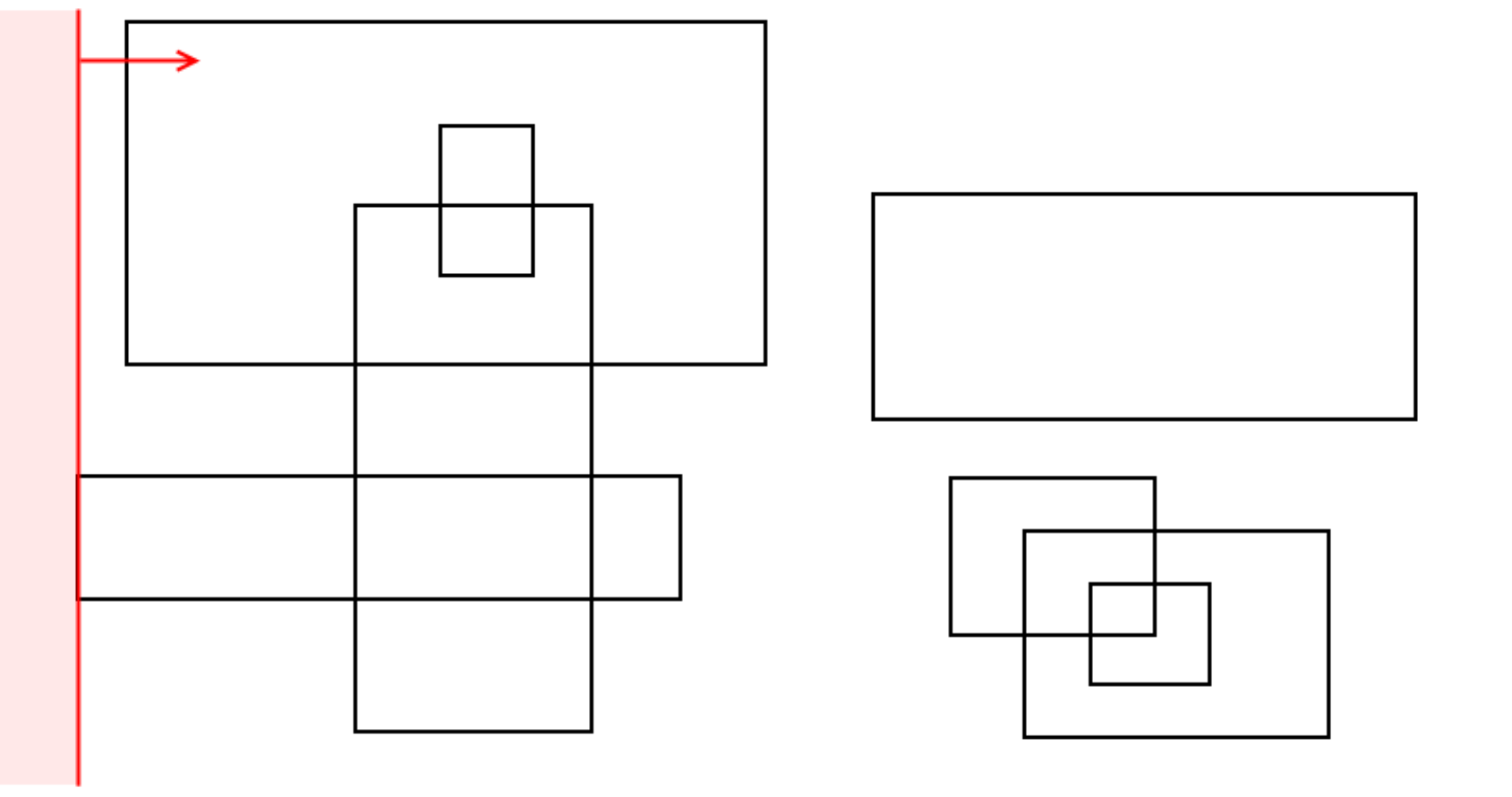
Sweep down to up to get length of vertical segments.

Idea is same as rectangle area union except:

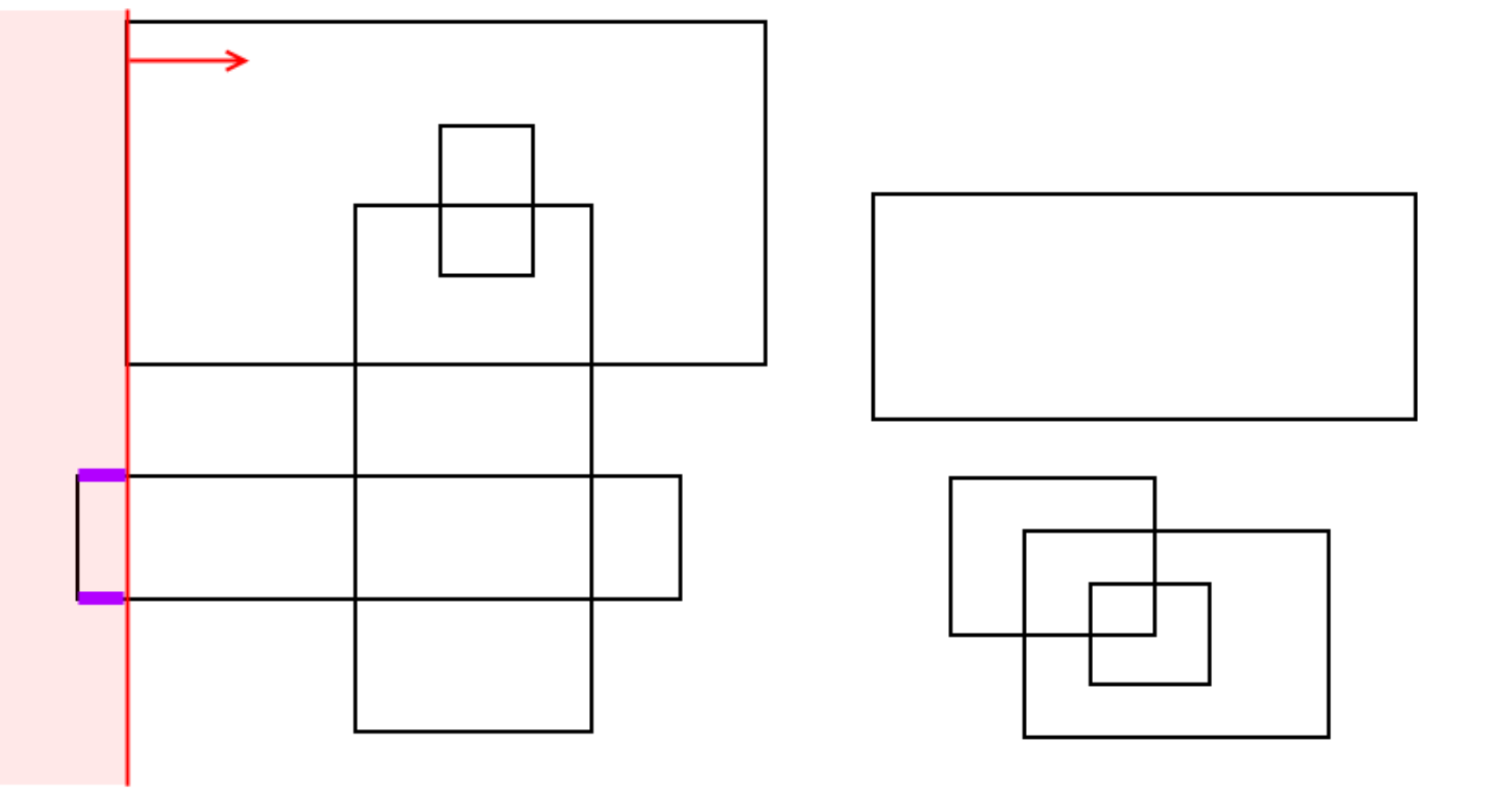
- Before processing events at $x$, add to perimeter $\Delta x \cdot$(#end points in segment union)

Time complexity: $O(n \log n)$

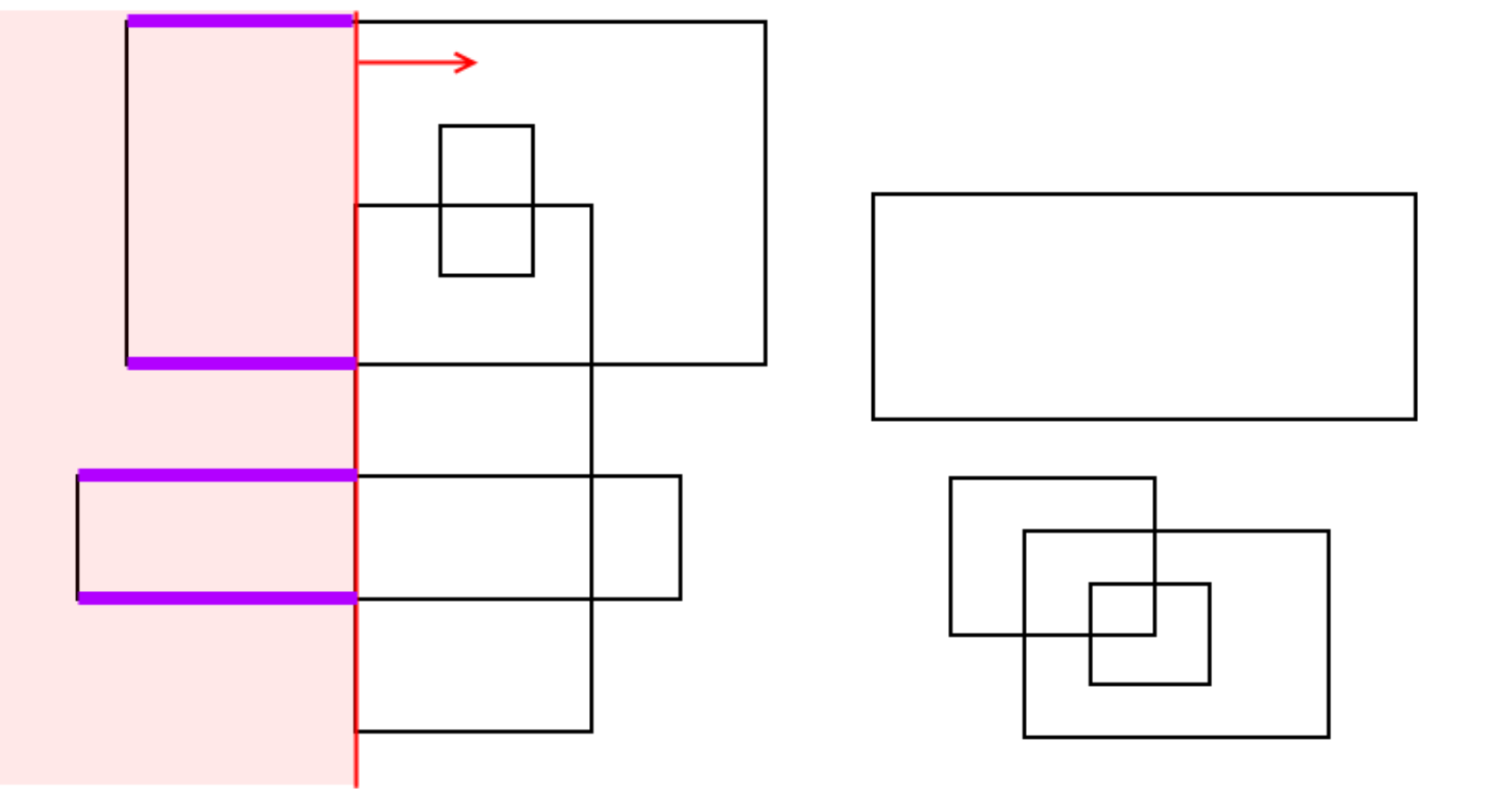
# Problem 3: Solution



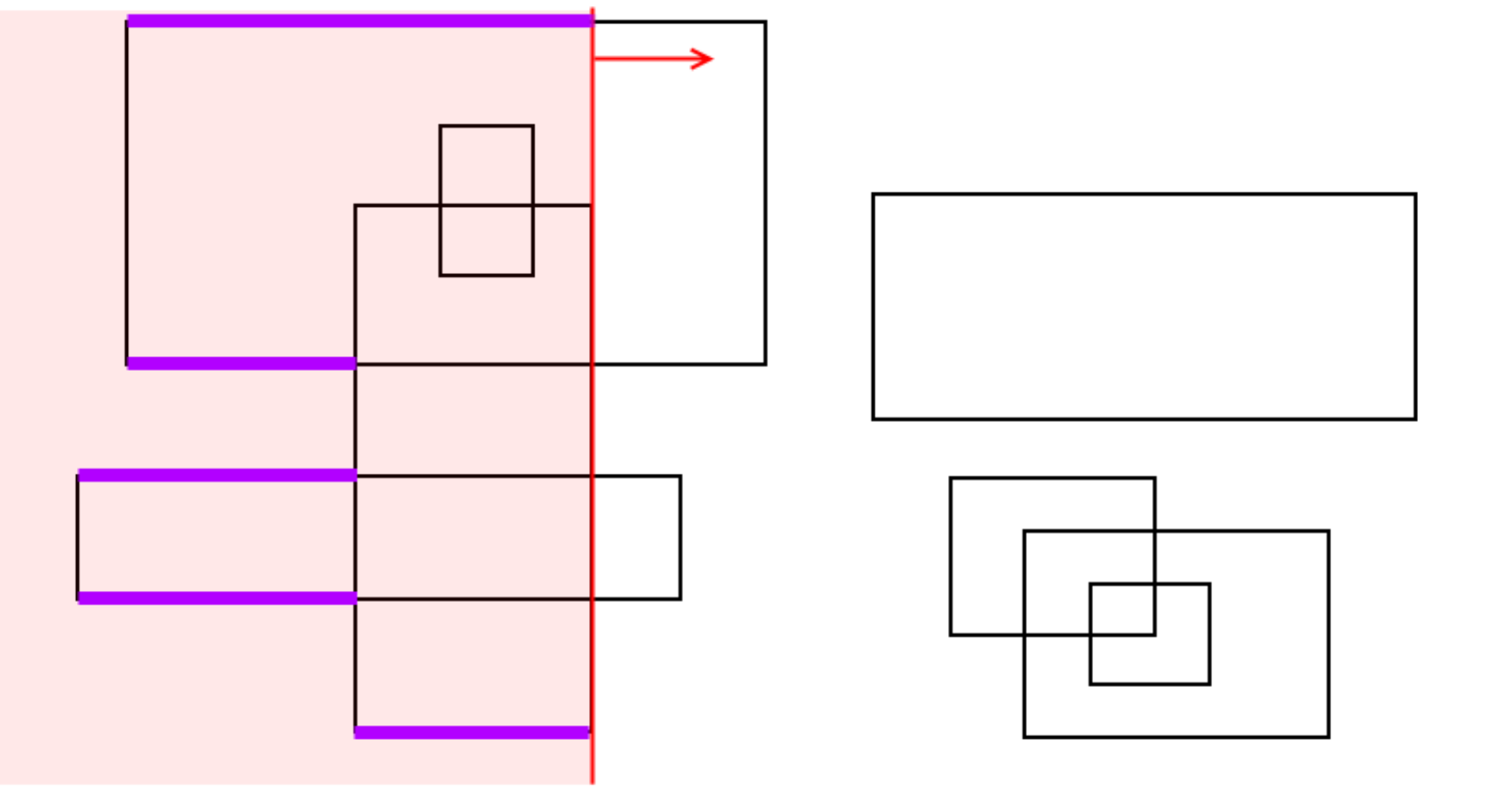
# Problem 3: Solution



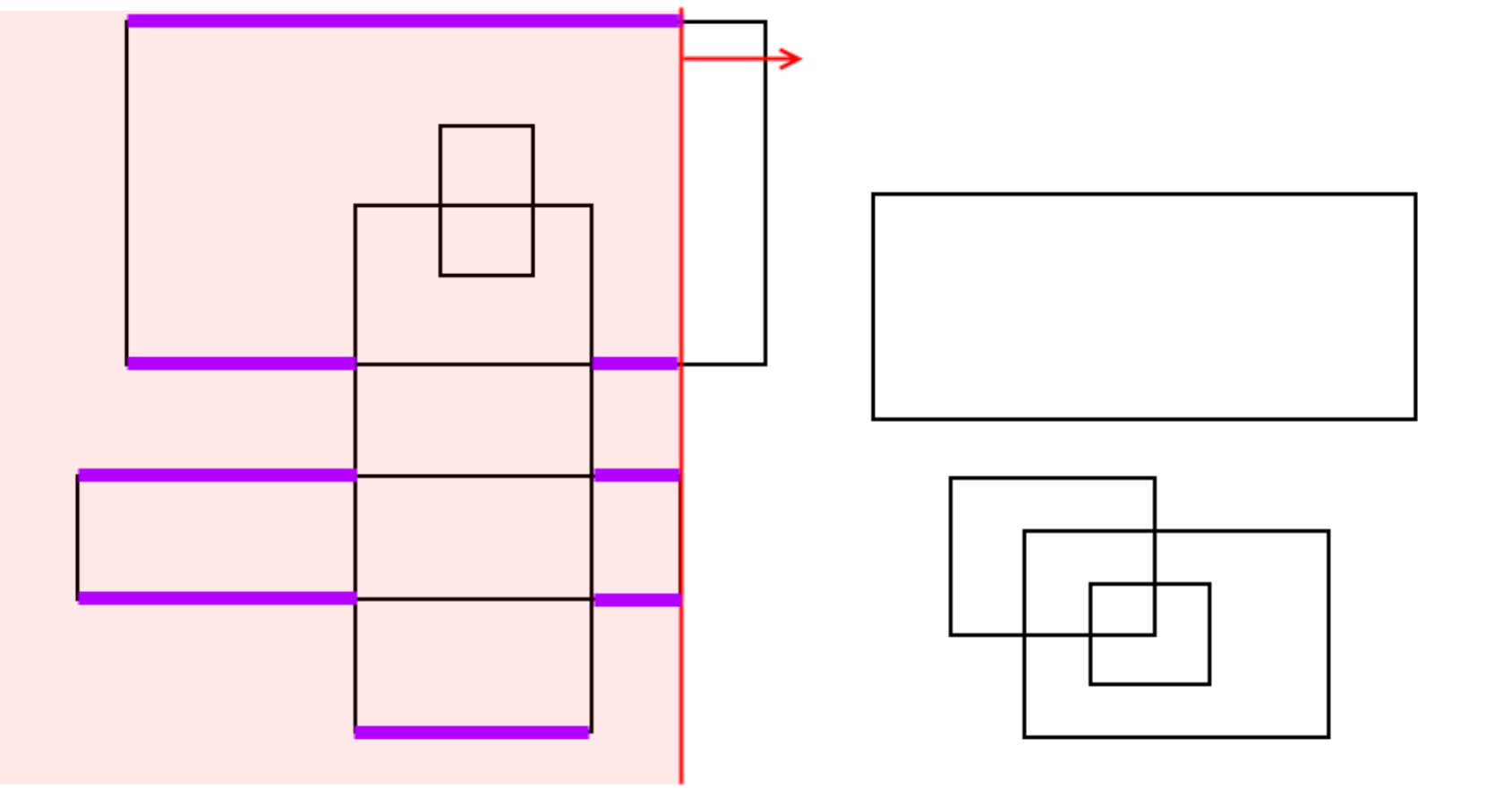
# Problem 3: Solution



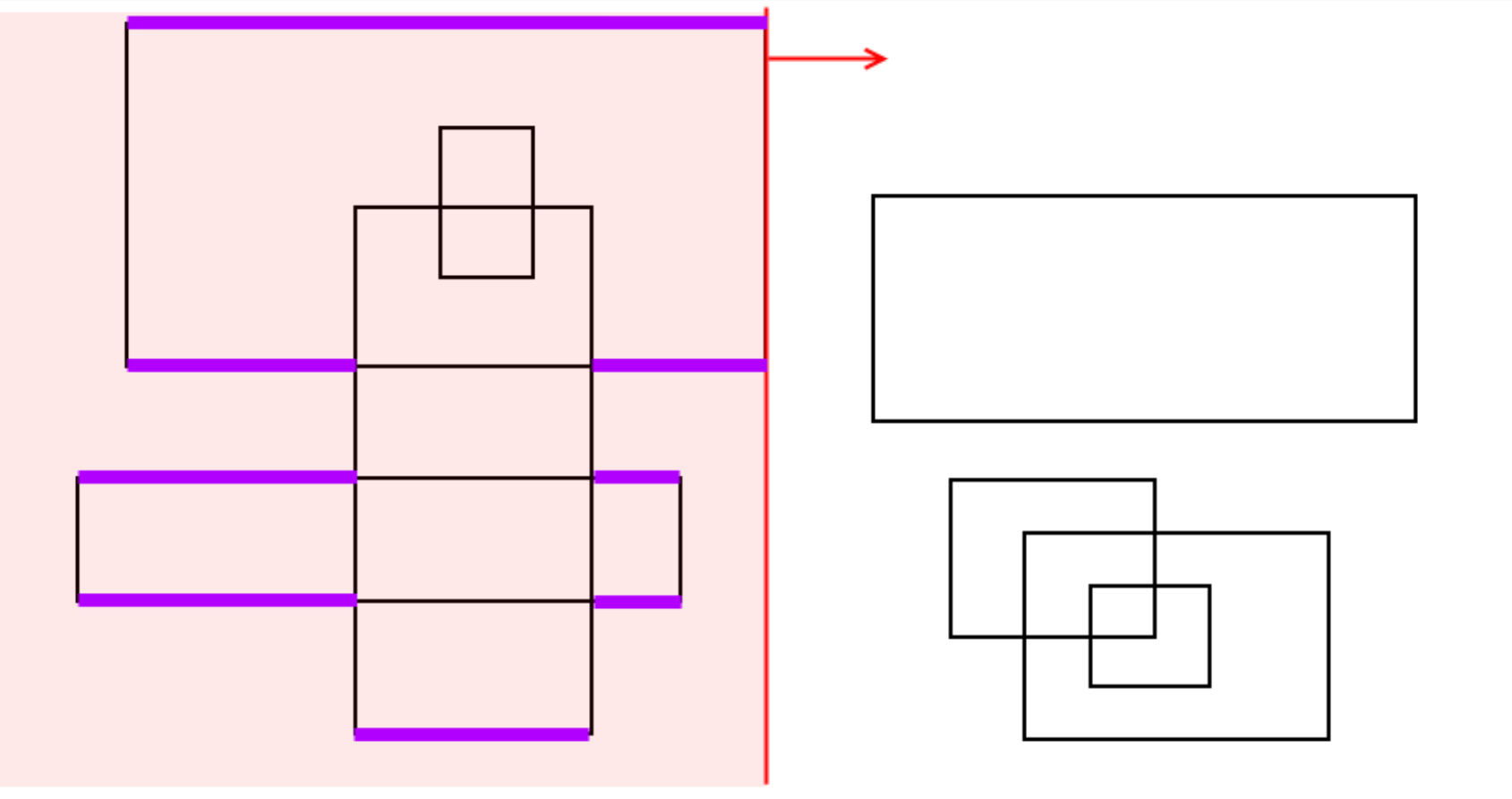
# Problem 3: Solution



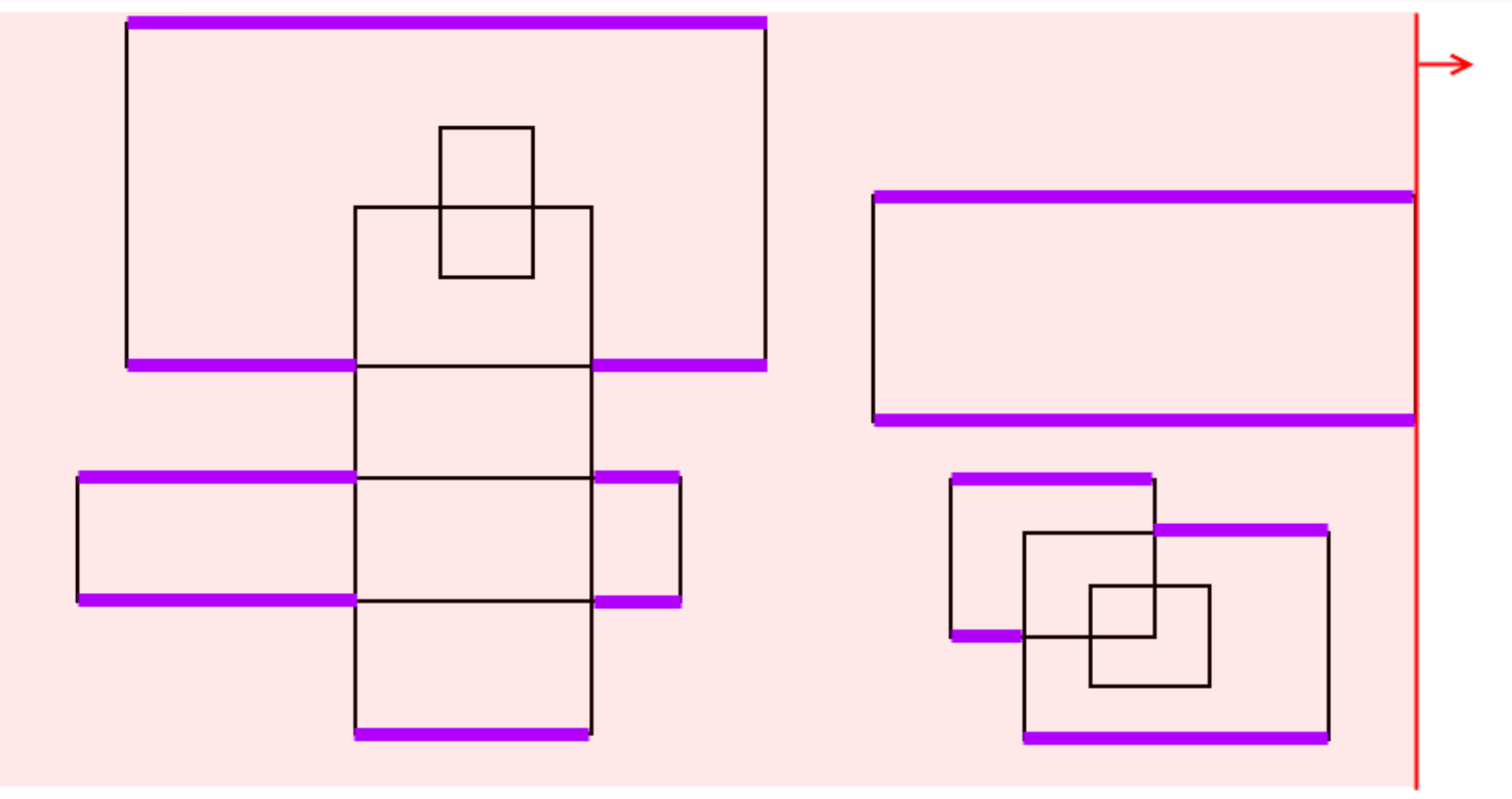
# Problem 3: Solution



# Problem 3: Solution

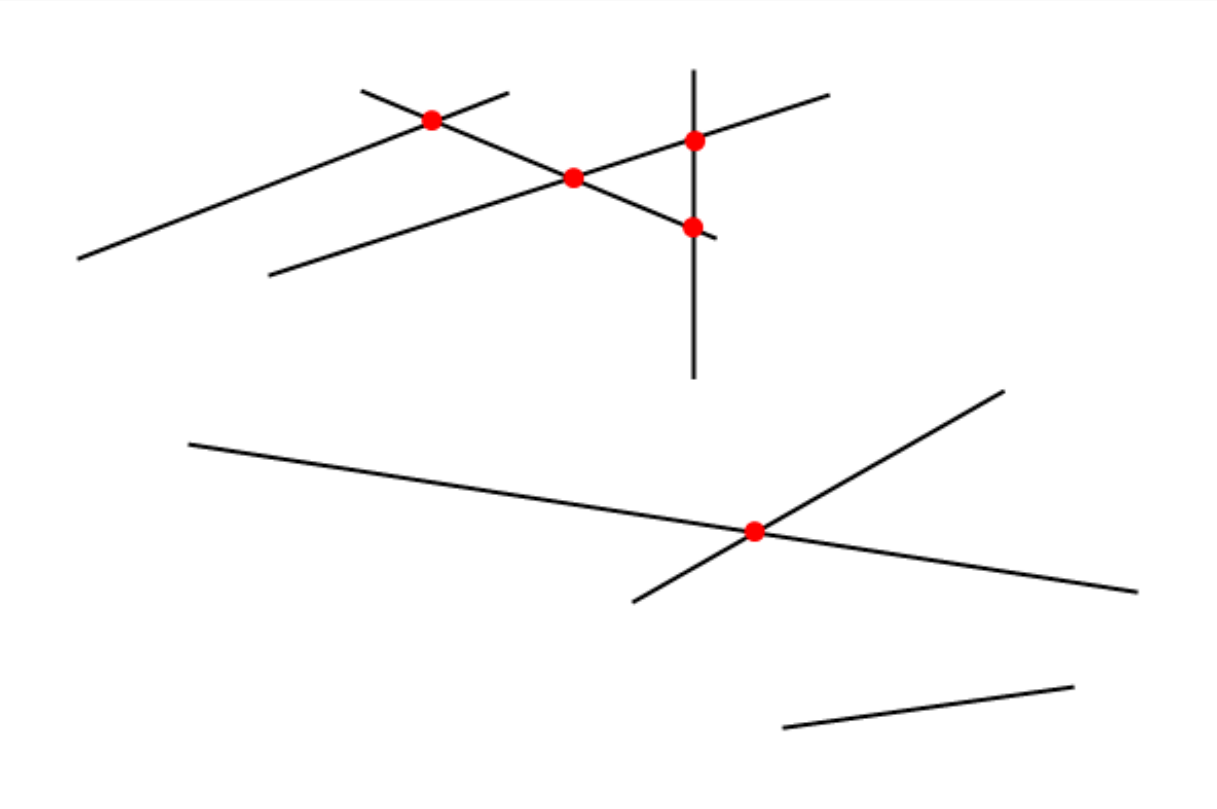


# Problem 3: Solution



# Problem 4: Line Segment Intersection

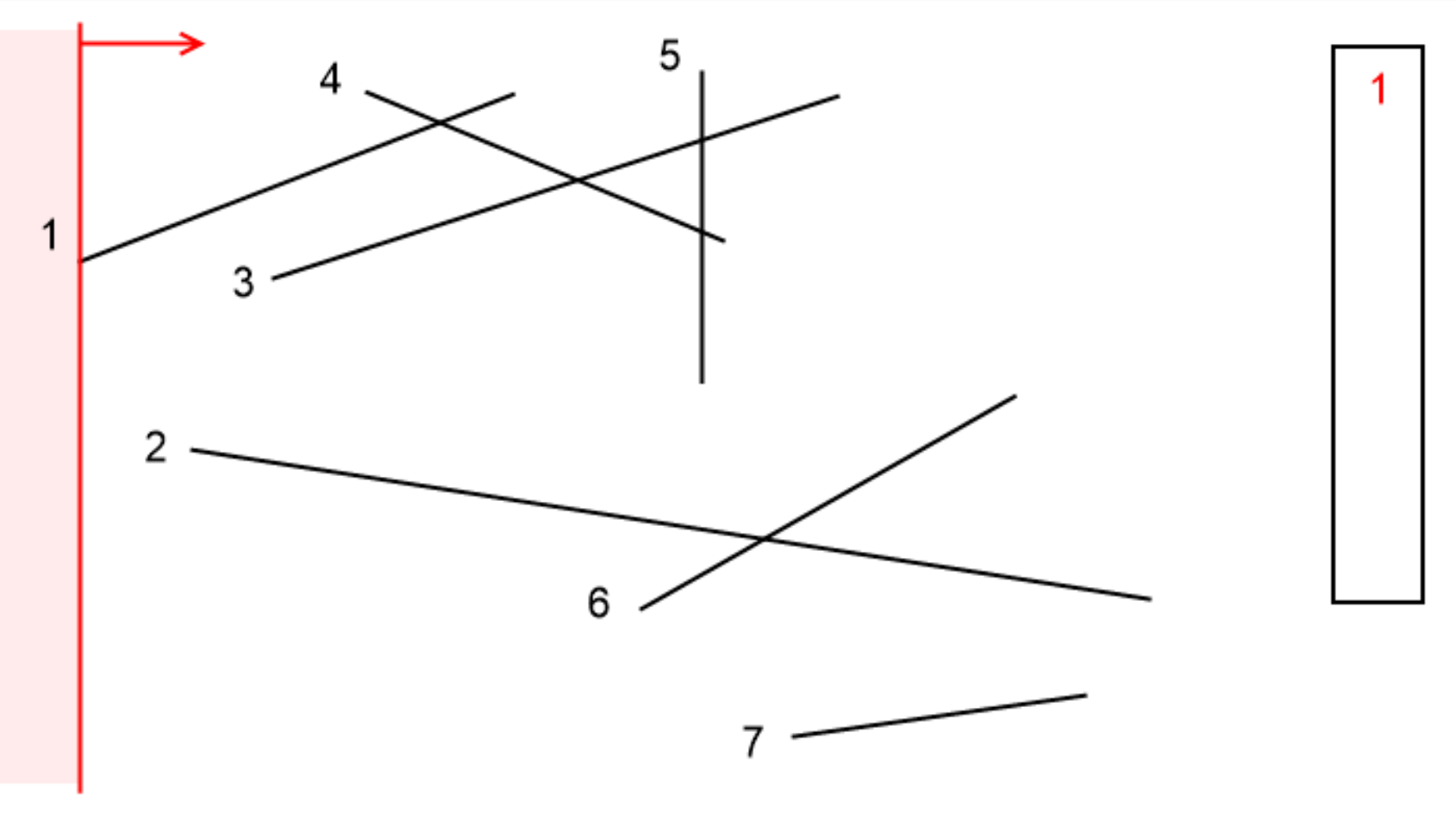
Find all intersections of these line segments!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

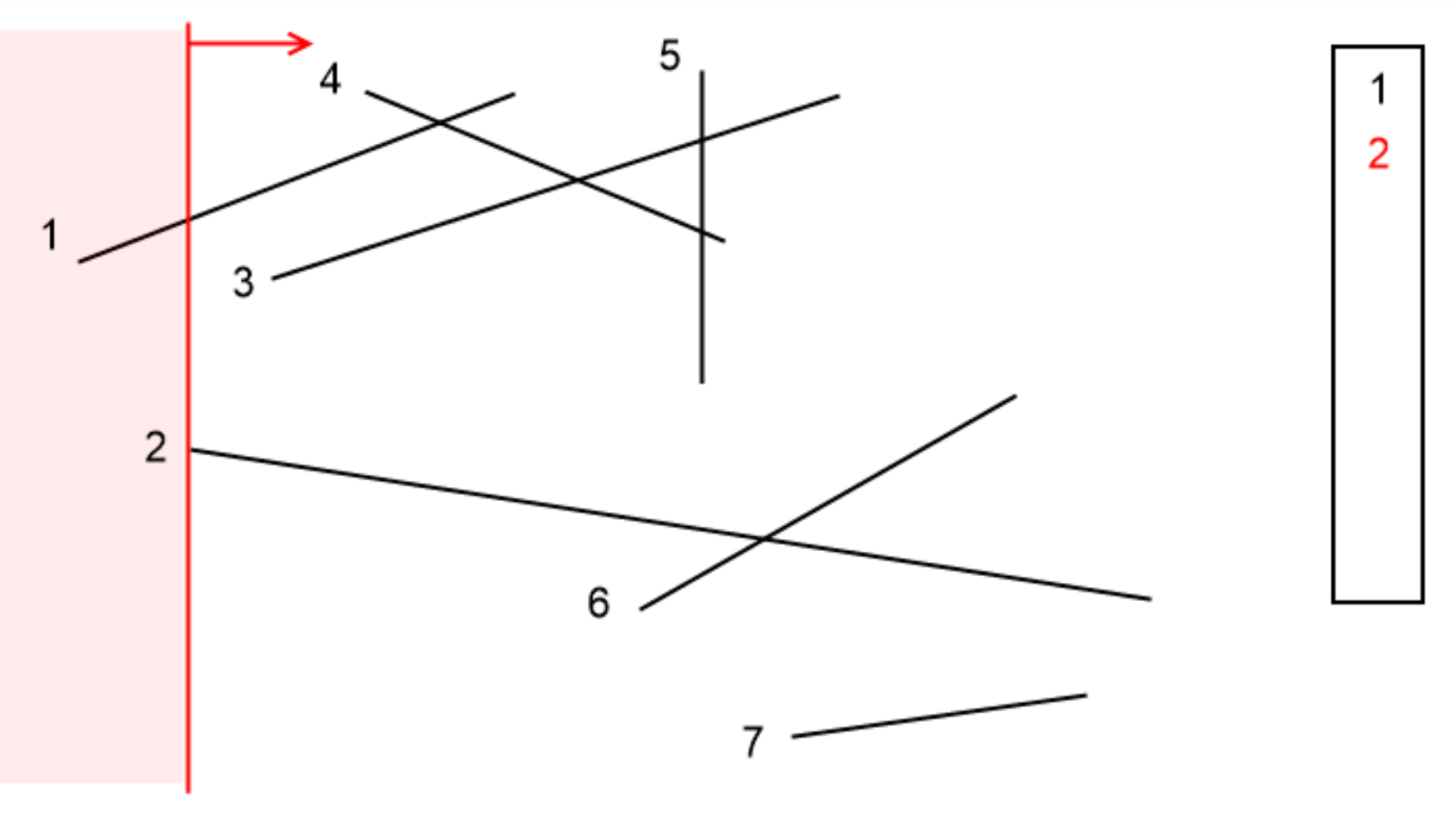
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

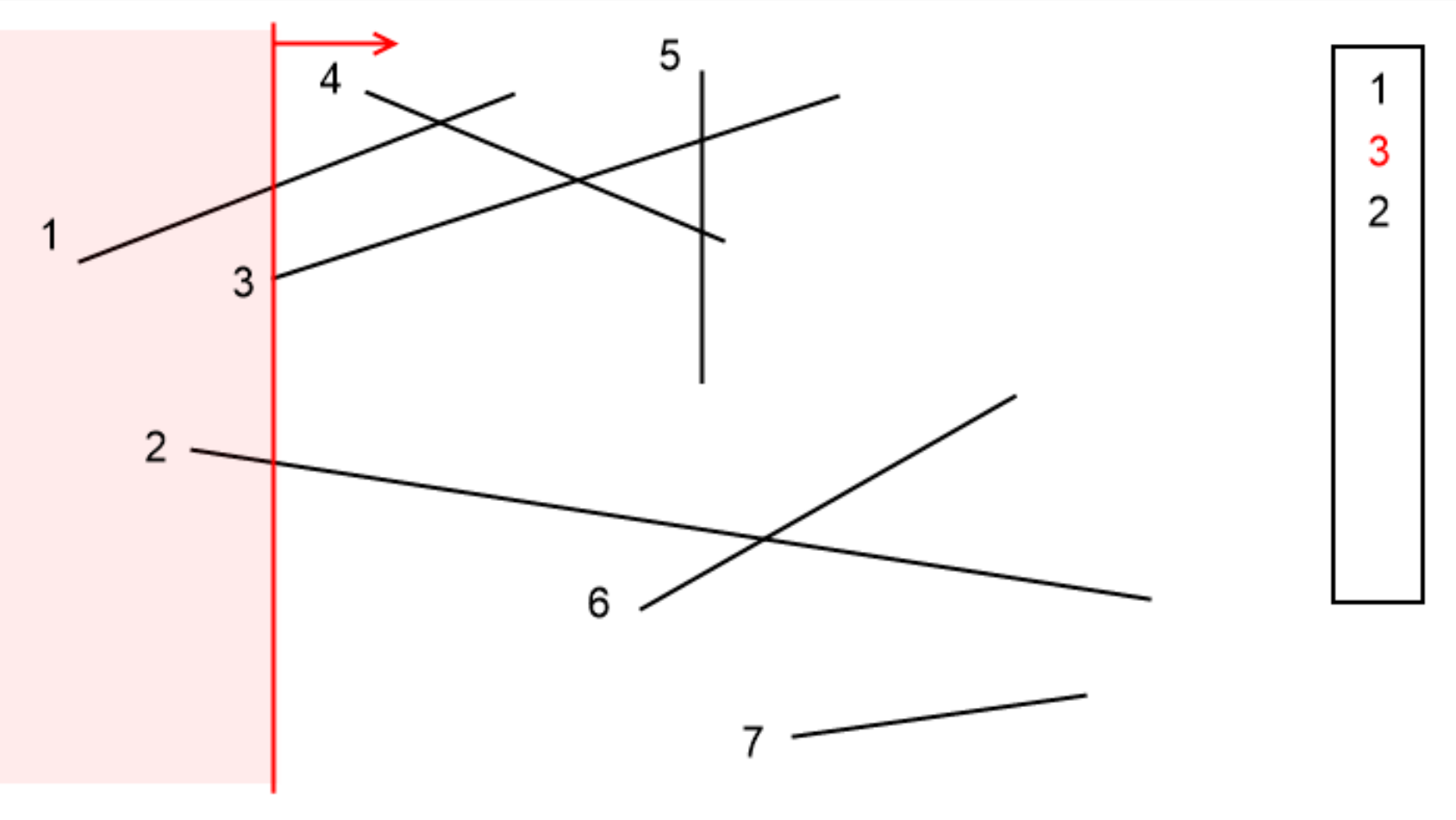
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

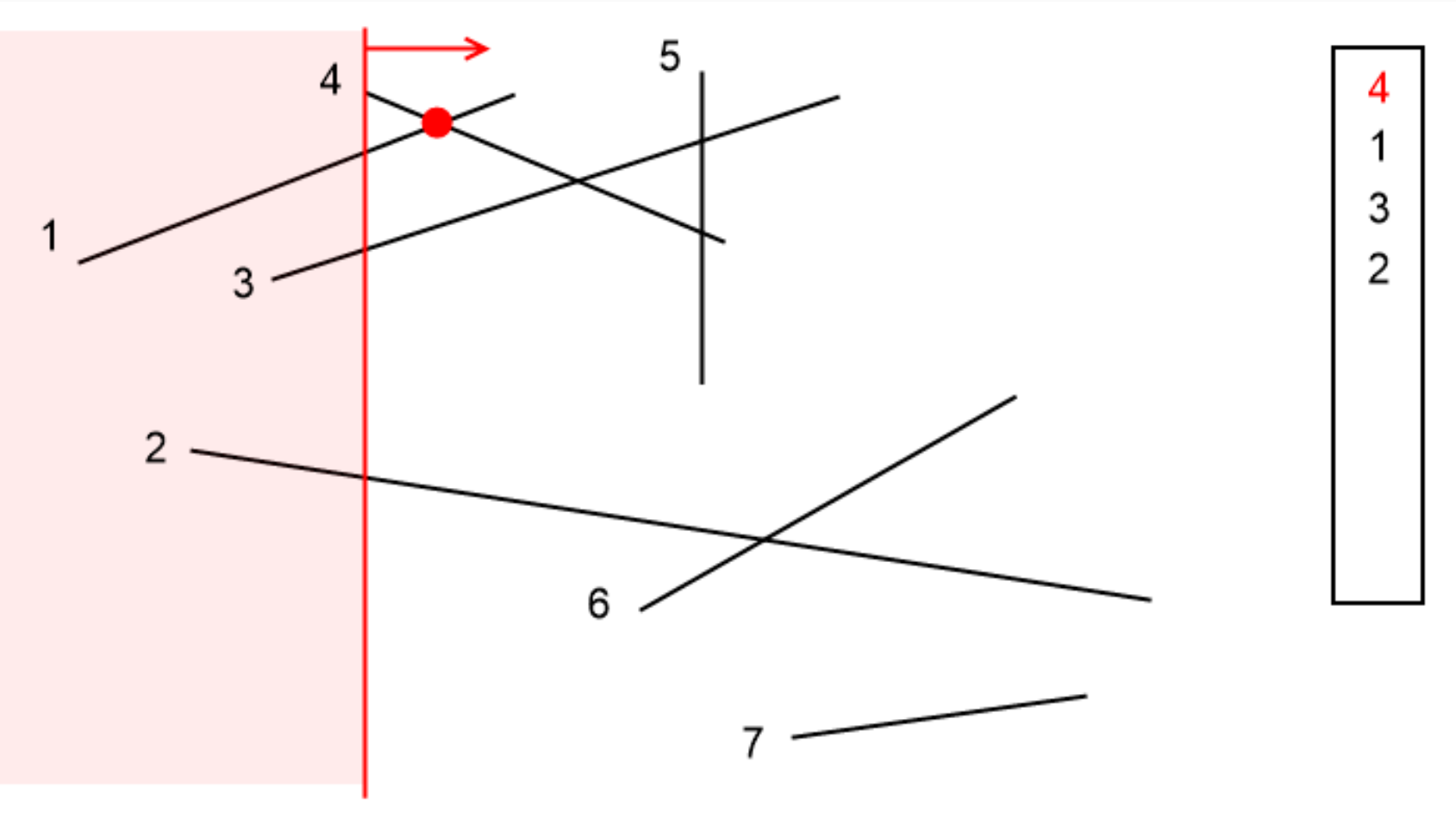
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

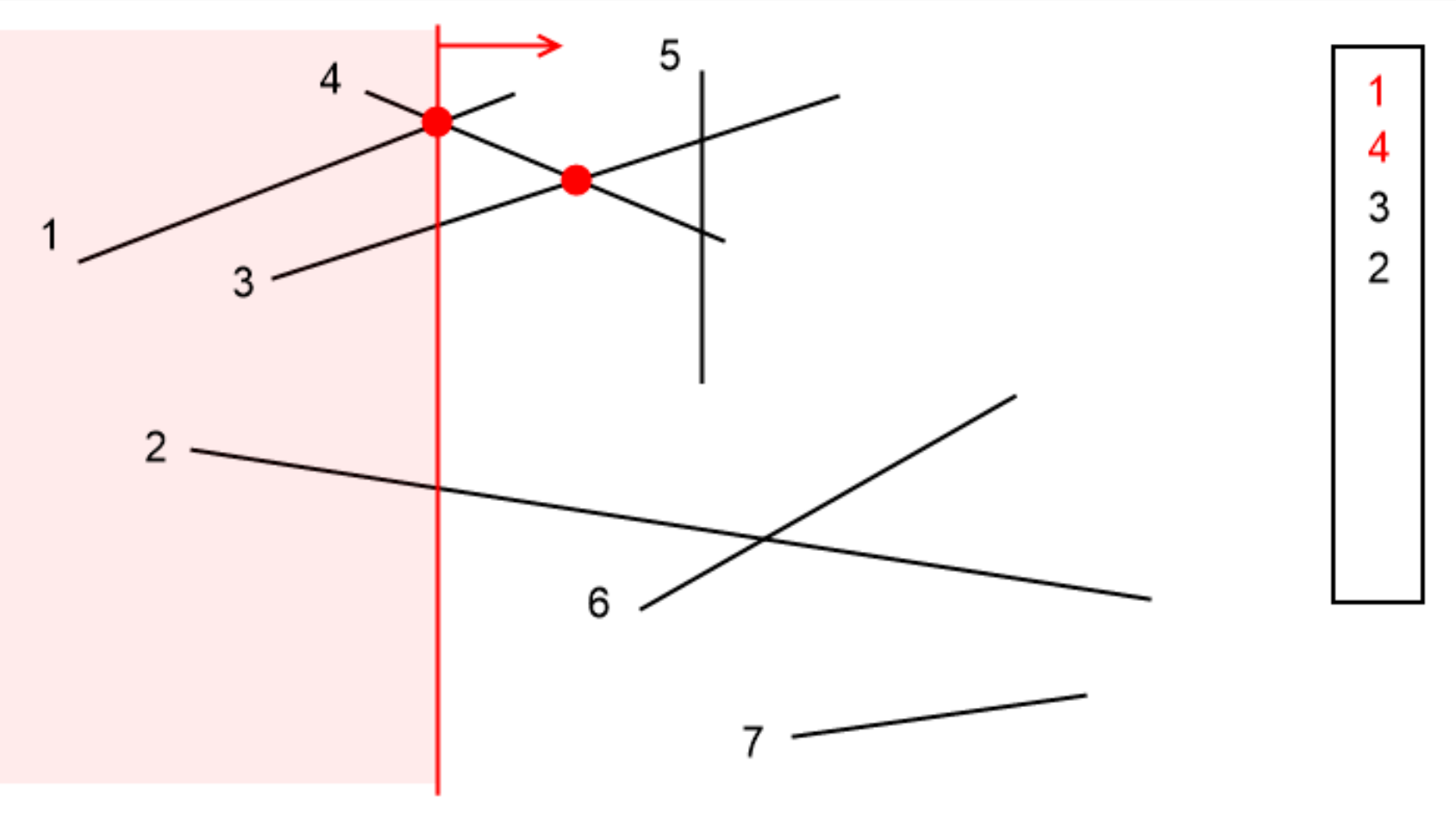
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

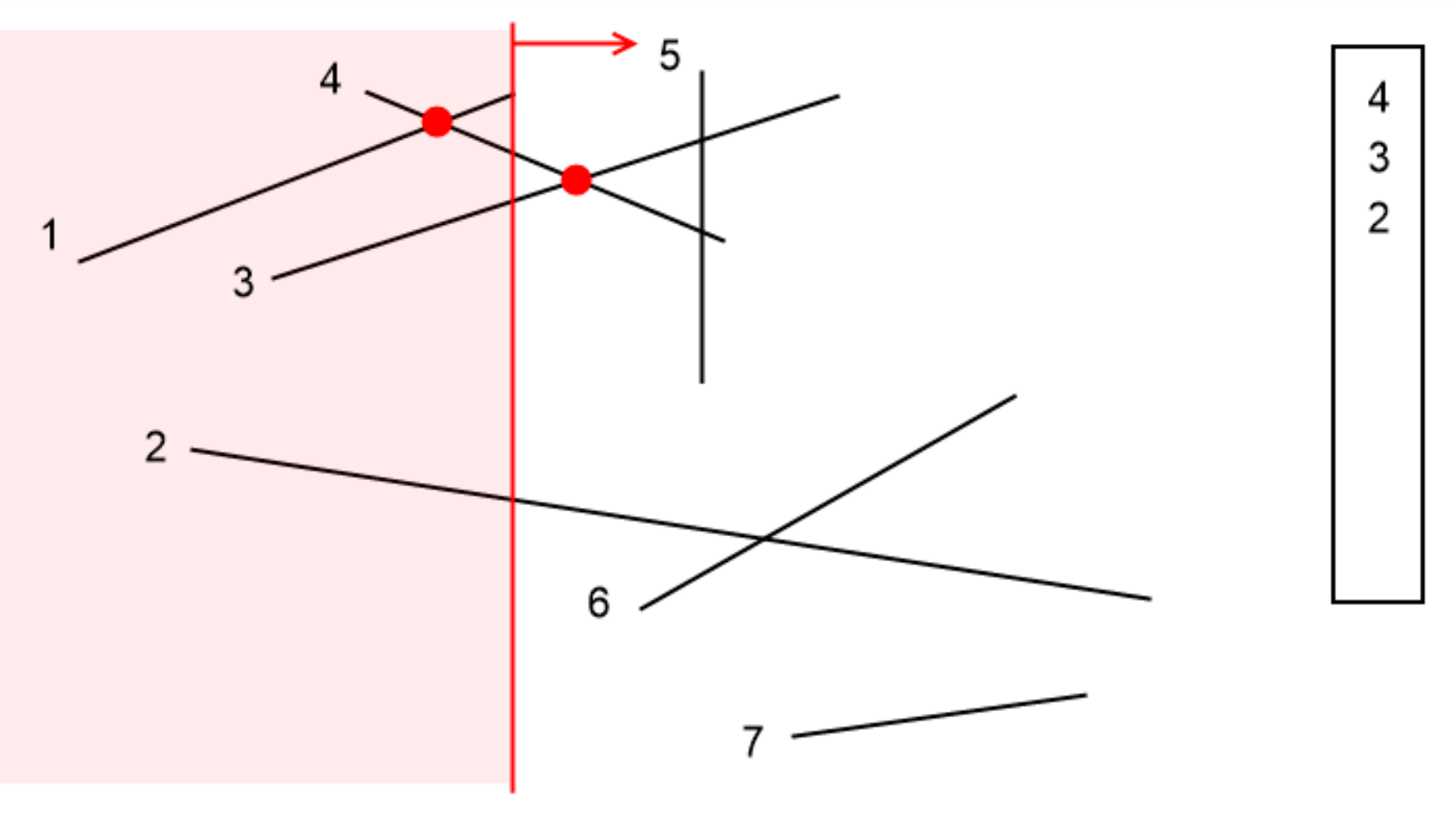
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

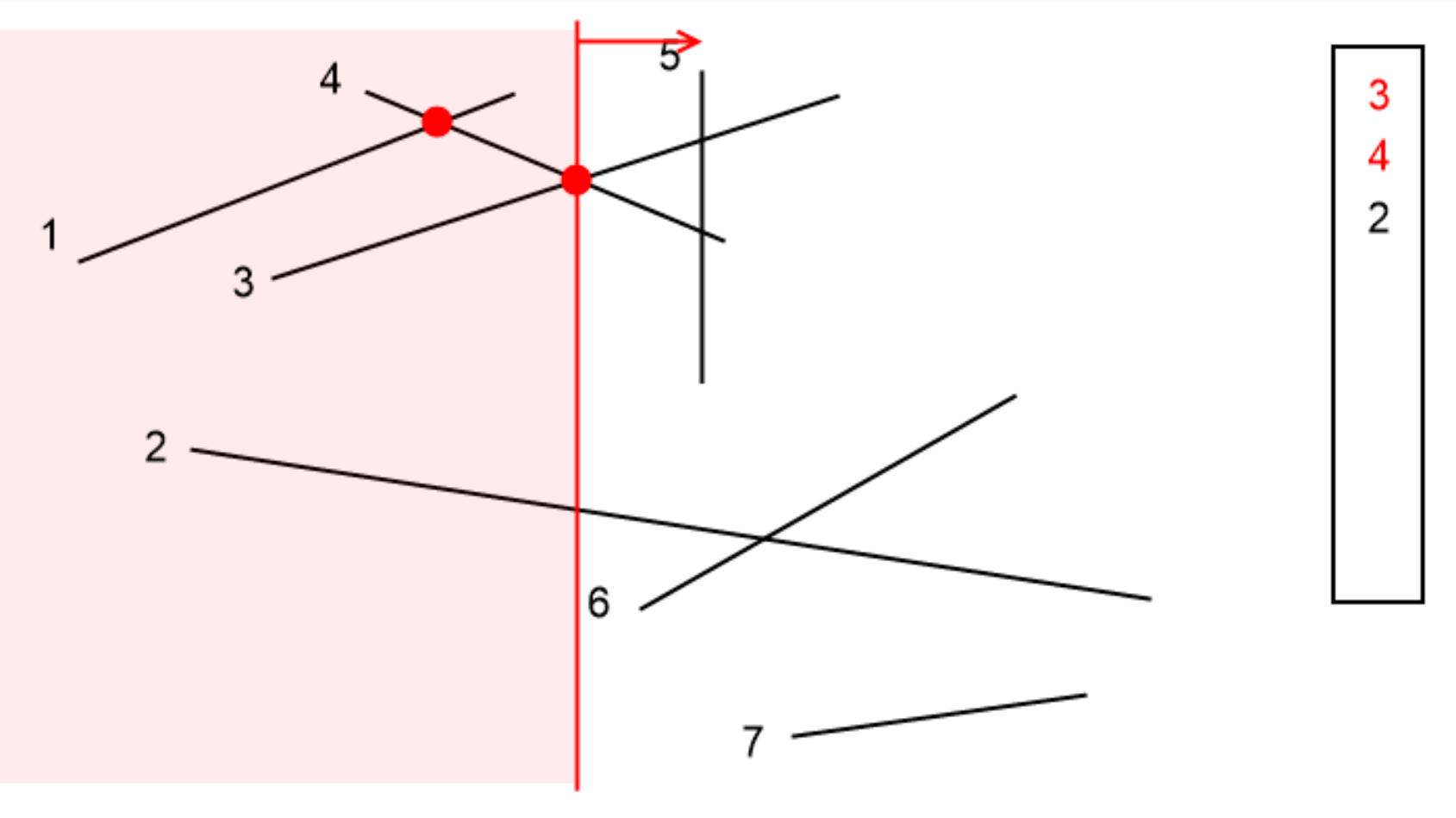
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

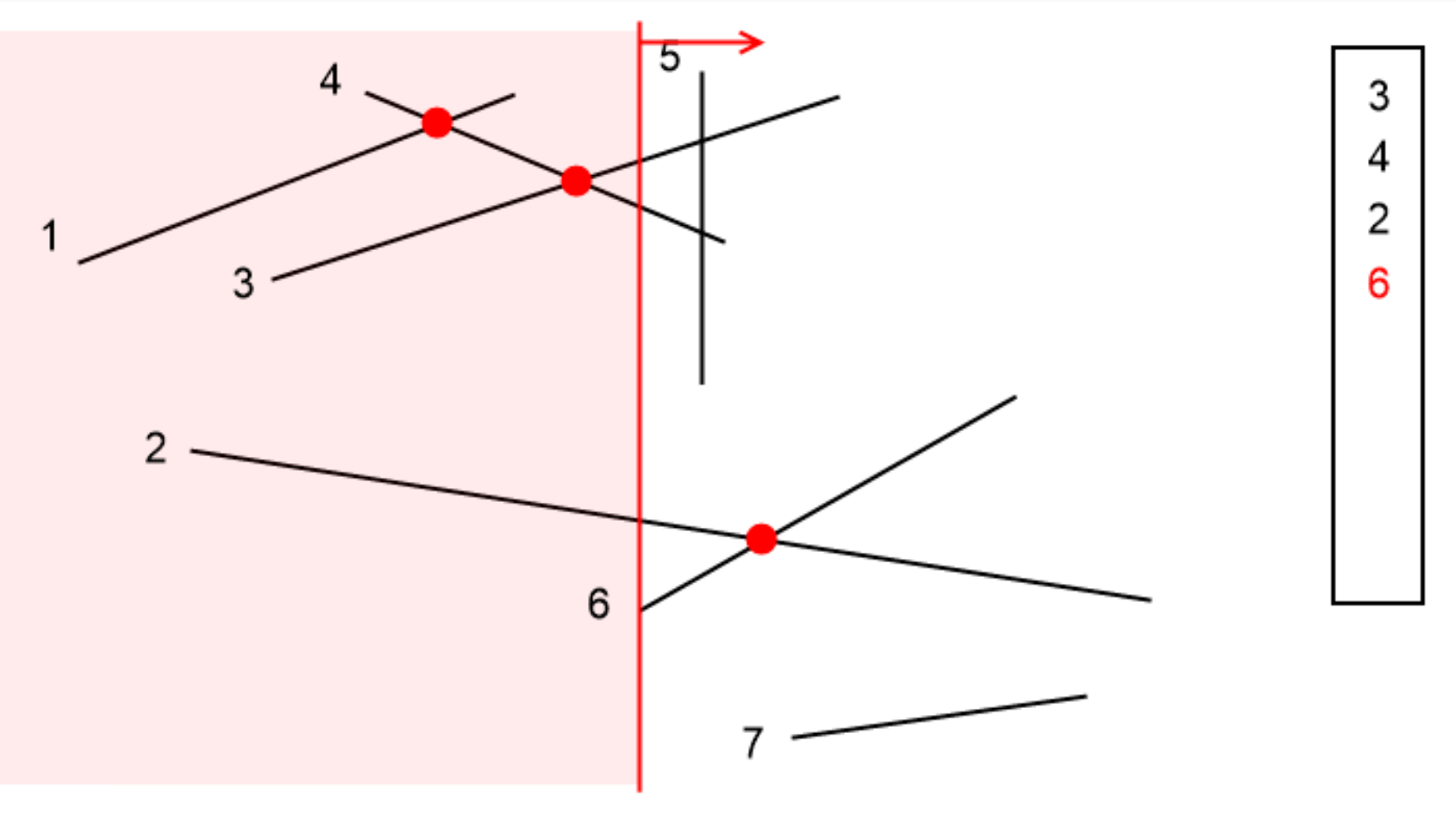
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

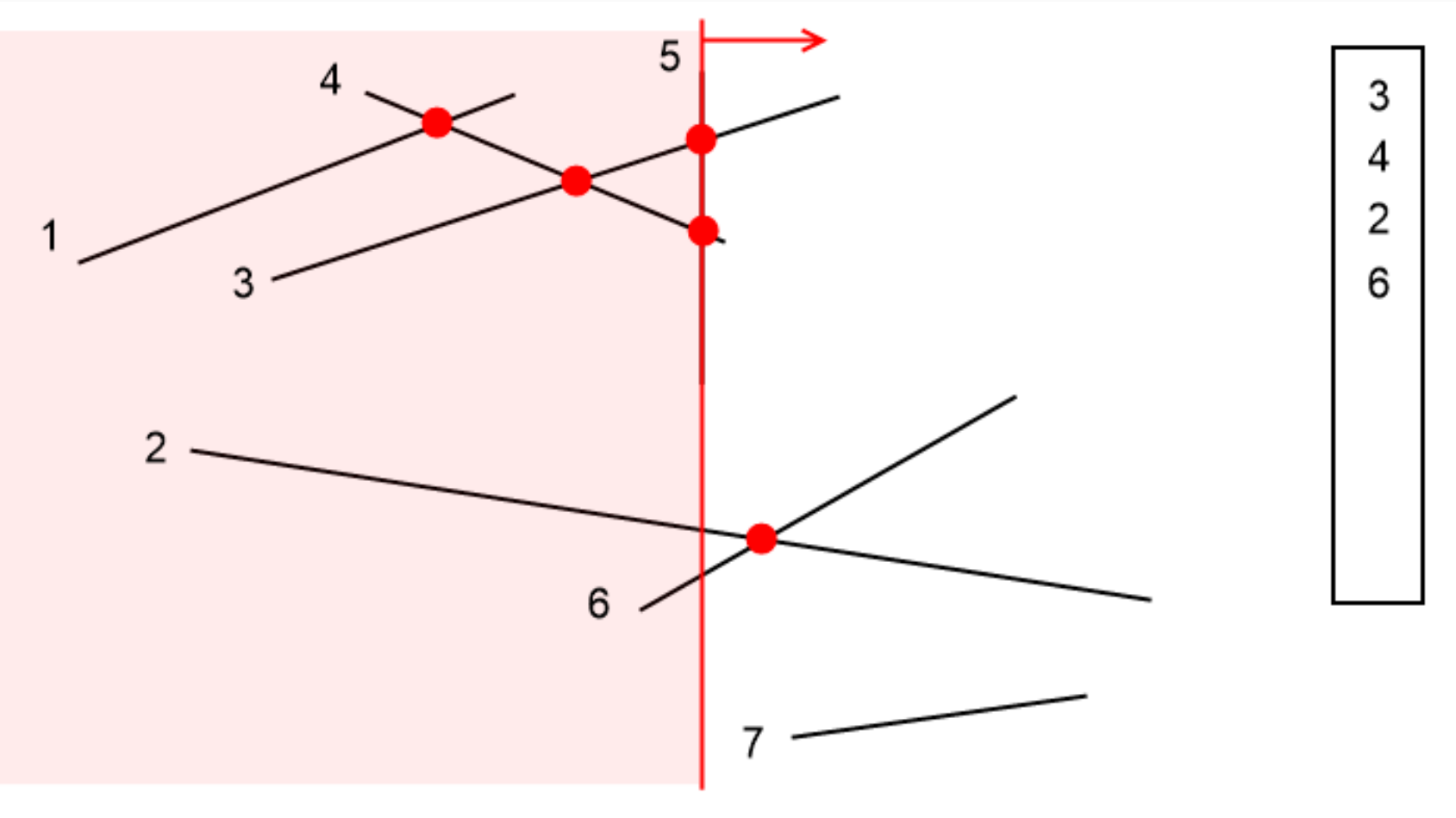
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

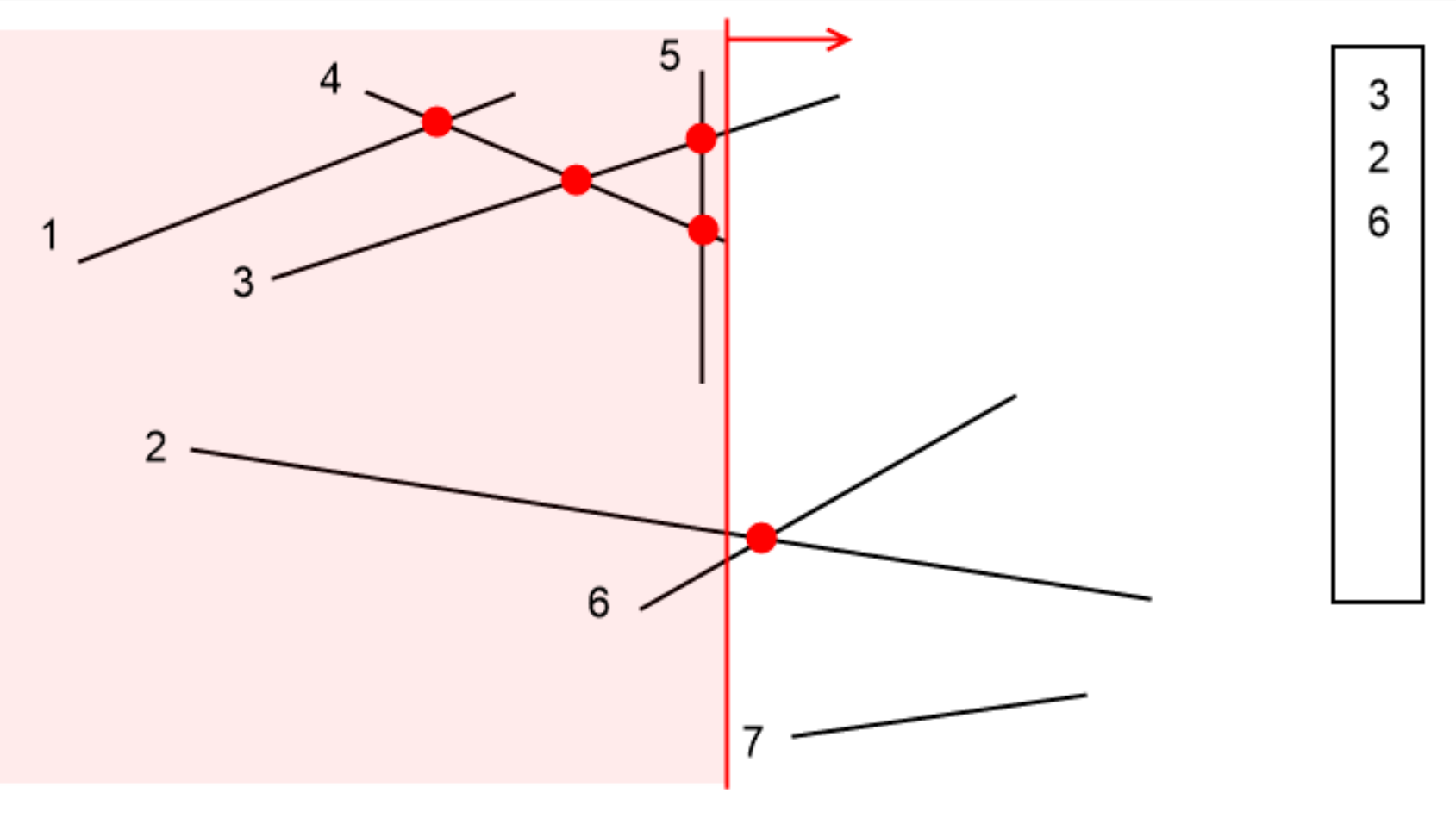
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

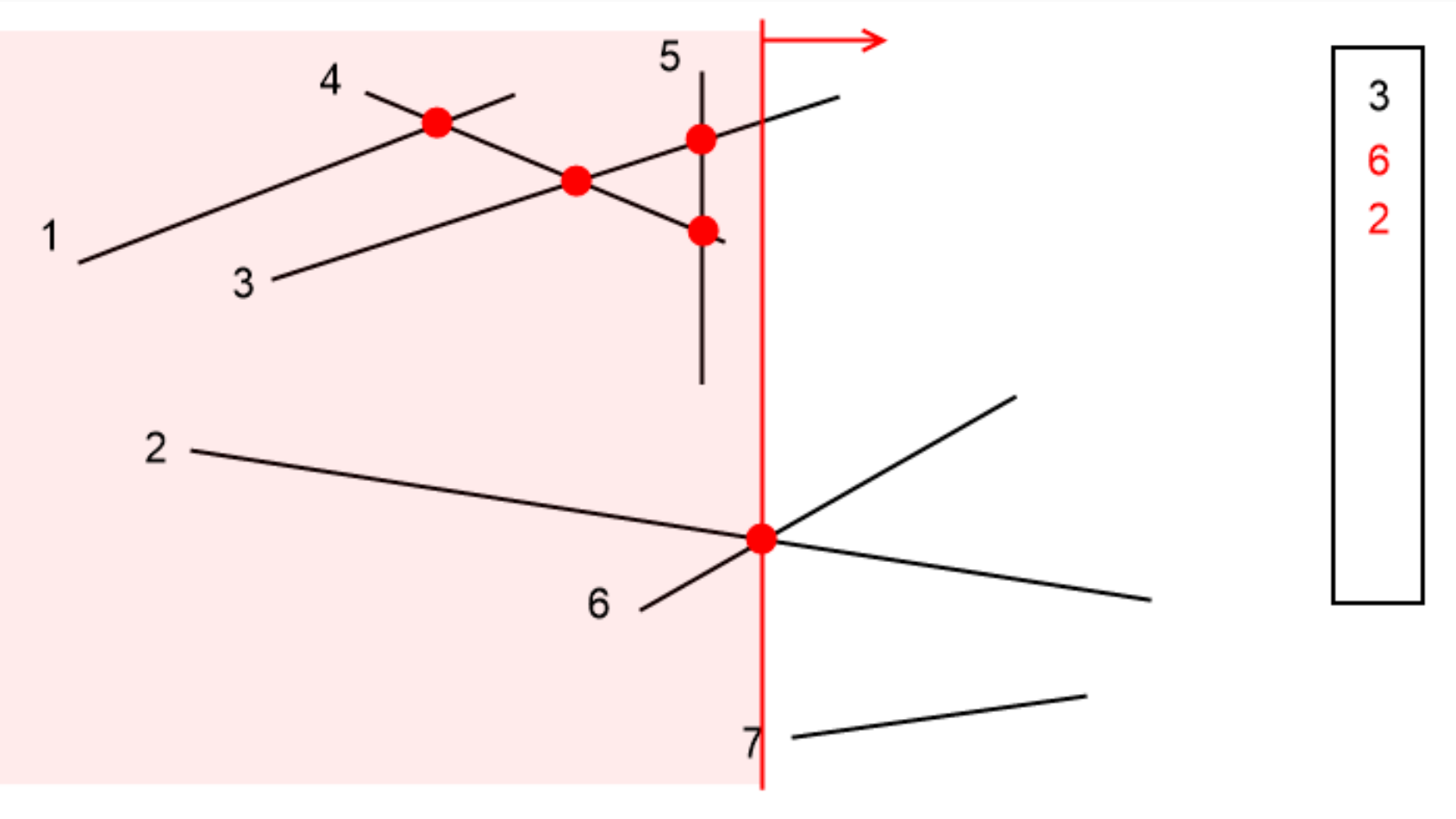
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

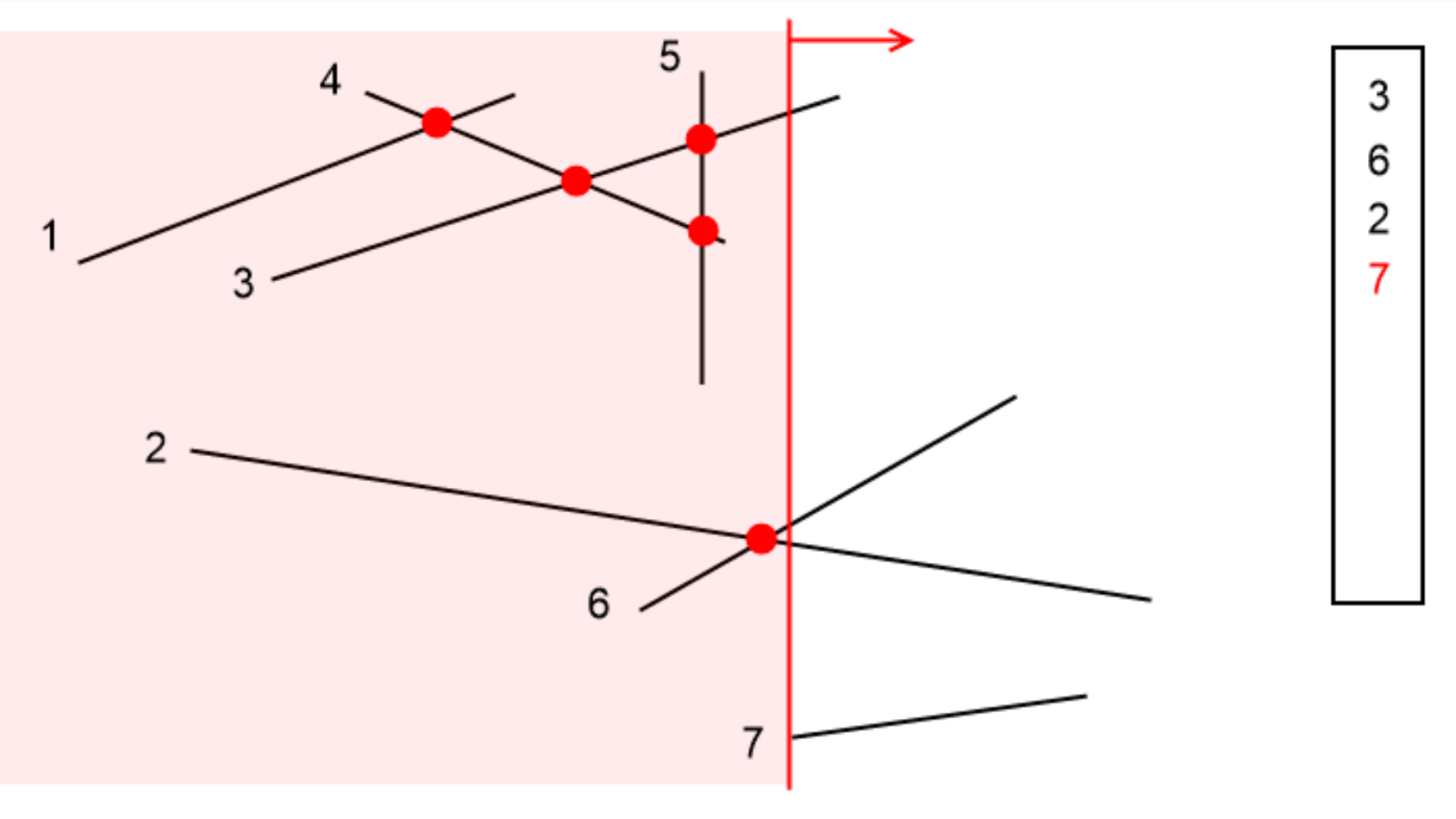
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

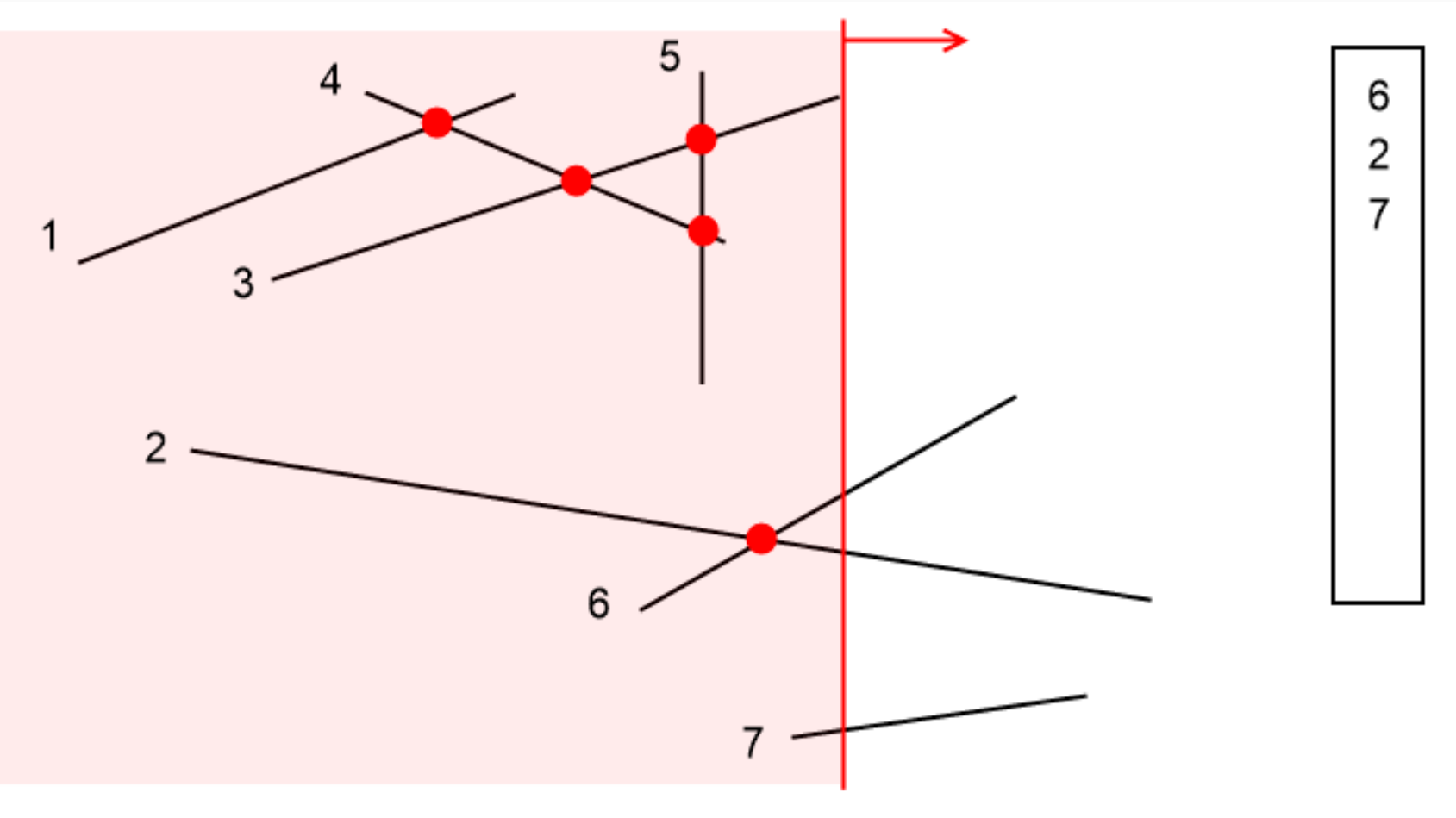
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

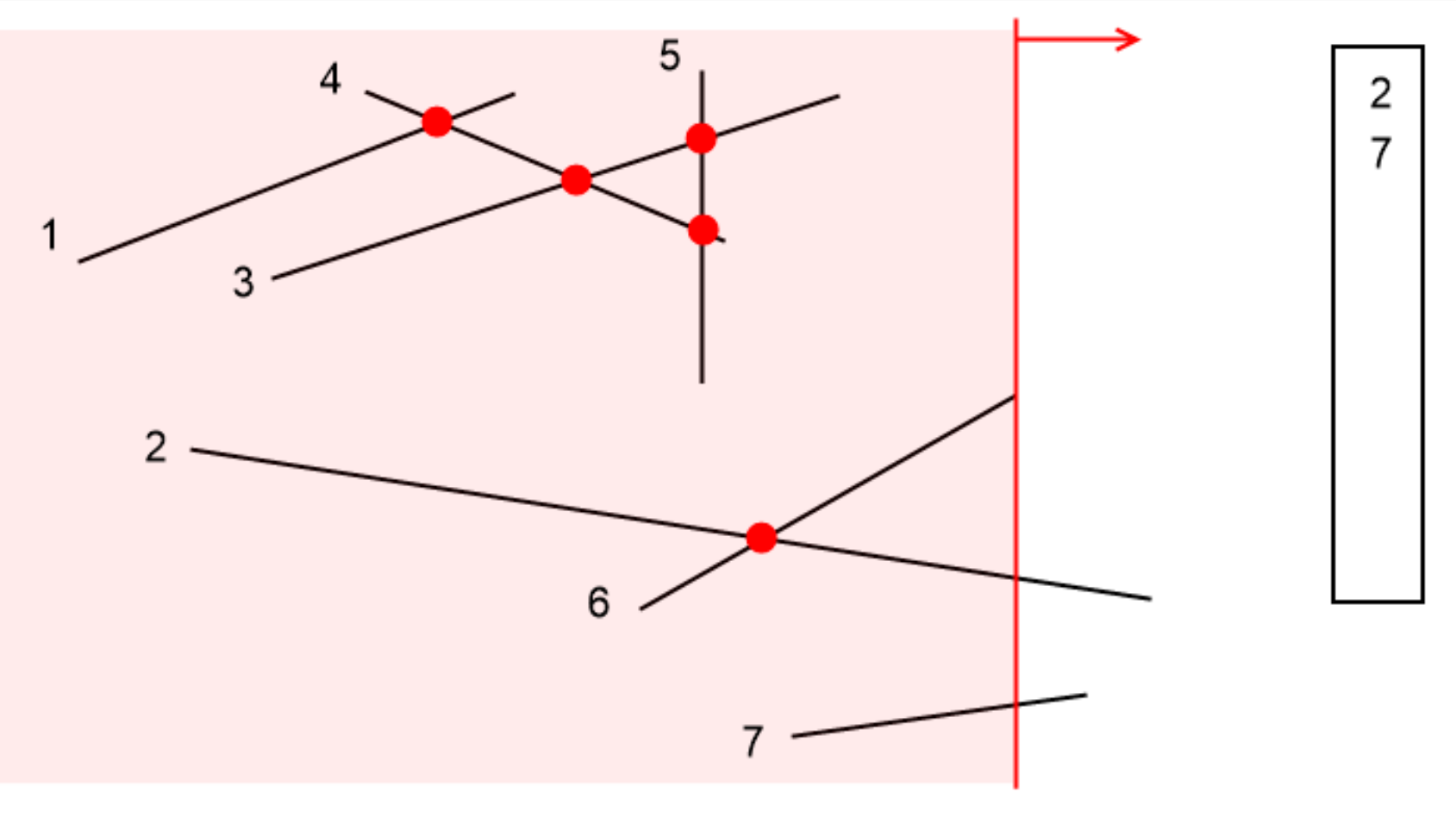
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

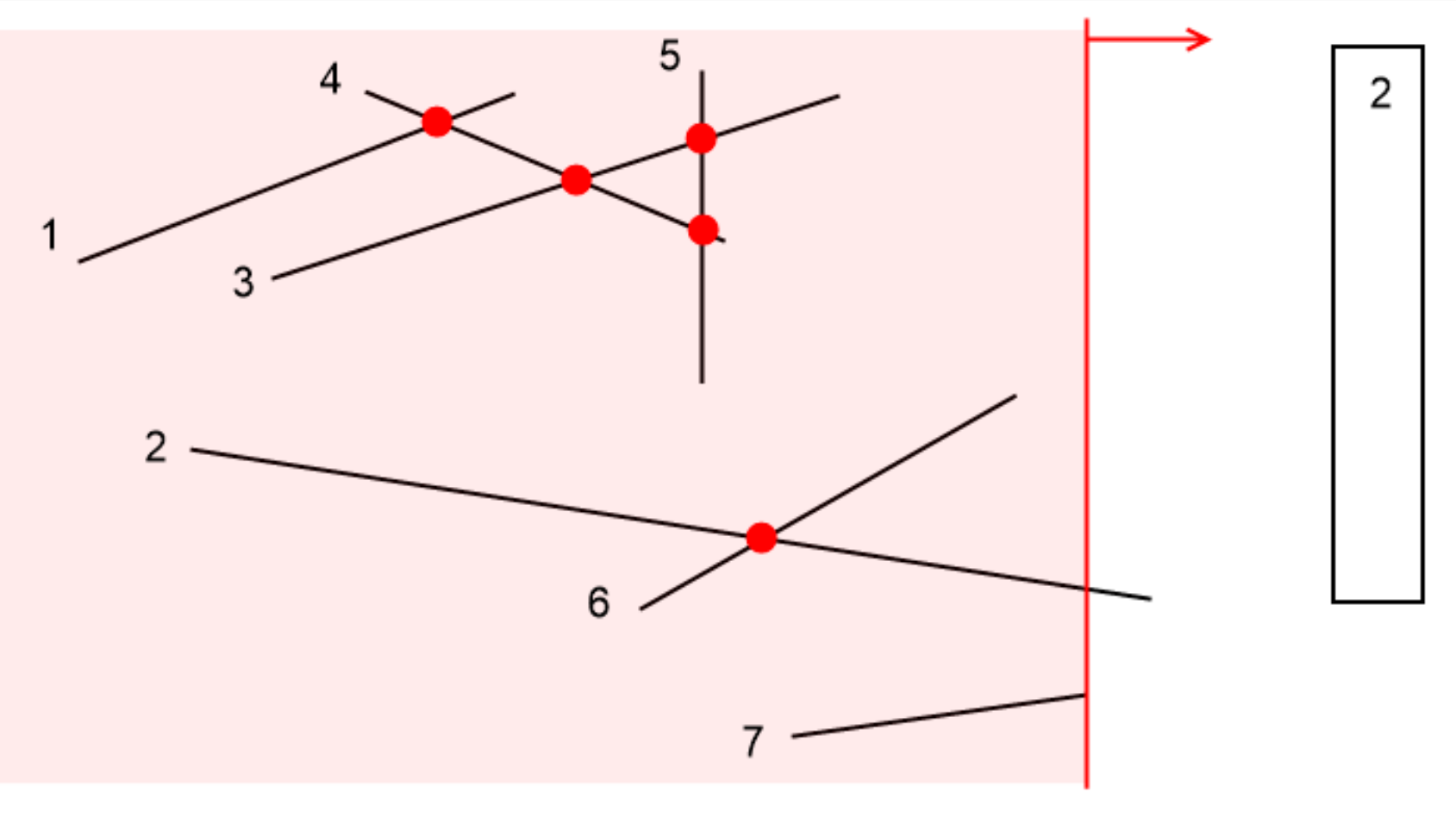
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

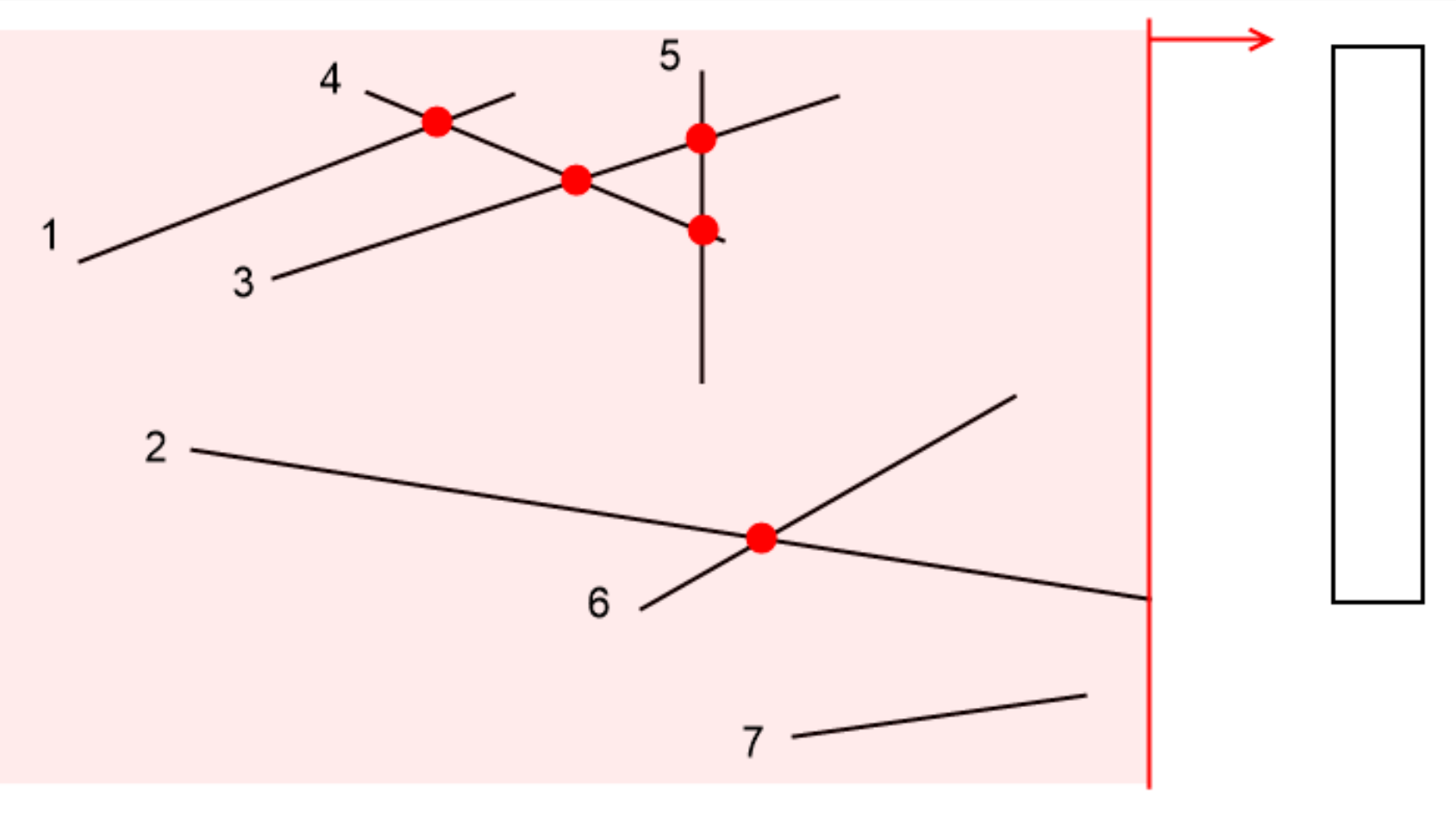
- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

Imagine sweeping a vertical line from left to right, observe that

- Vertical ordering only change on intersection!
- Intersection must be between two adjacent segs in vert. order!



# Problem 4: Solution

State: set of line segments crossing the sweep line, ordered by $y$ coordinate

Events: left end of segment, right end of segment, intersection, vertical line, put in a priority queue sorted by $x$

Actions

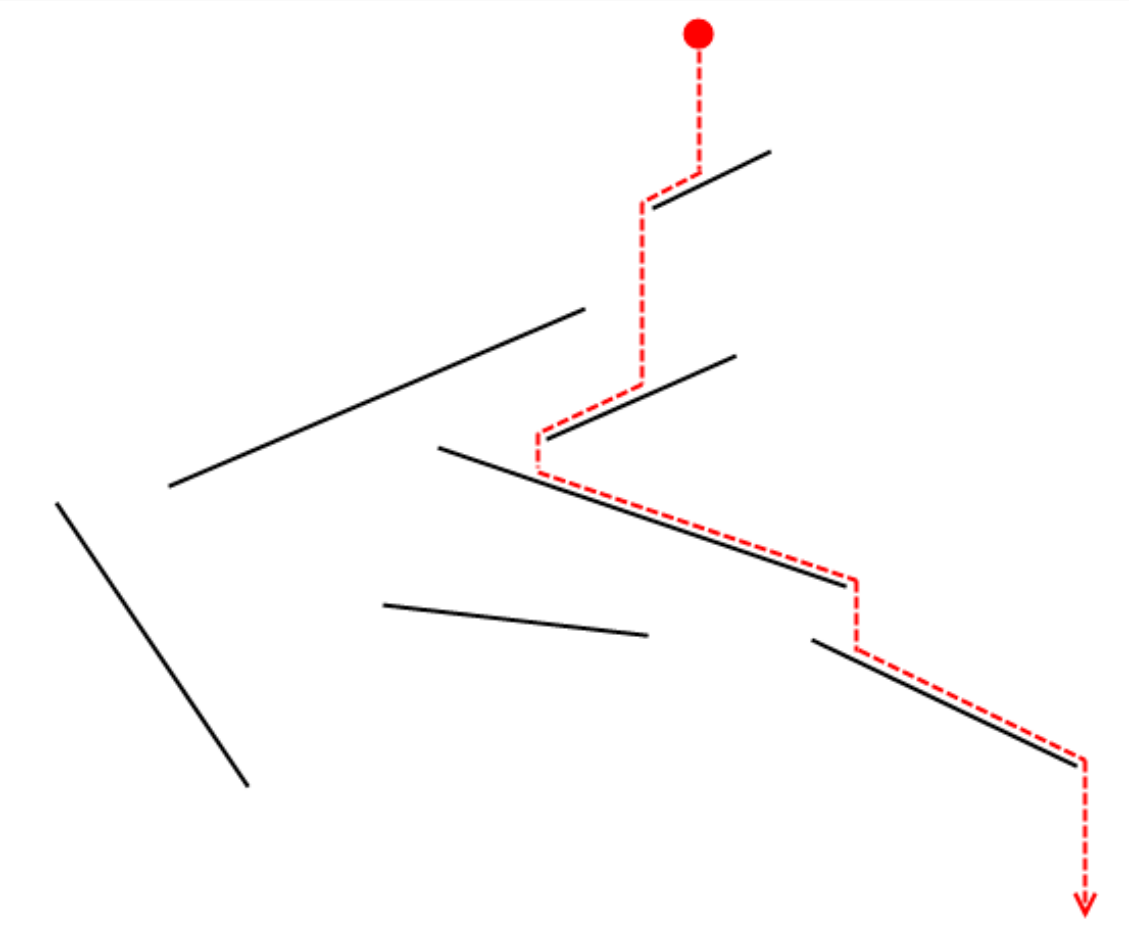
- Left end of segment $\rightarrow$ insert into set
- Right end of segment $\rightarrow$ remove from set
- Vertical segment $\rightarrow$ binary search to find segments it intersect
- Intersection $\rightarrow$ flip ordering of the two (or more) segments
- After every insert/remove, check intersection between segments that become adjacent

Time complexity: $O((n + q) \log n)$ where $q =$ number of intersections

Data structure needed: balanced binary search tree

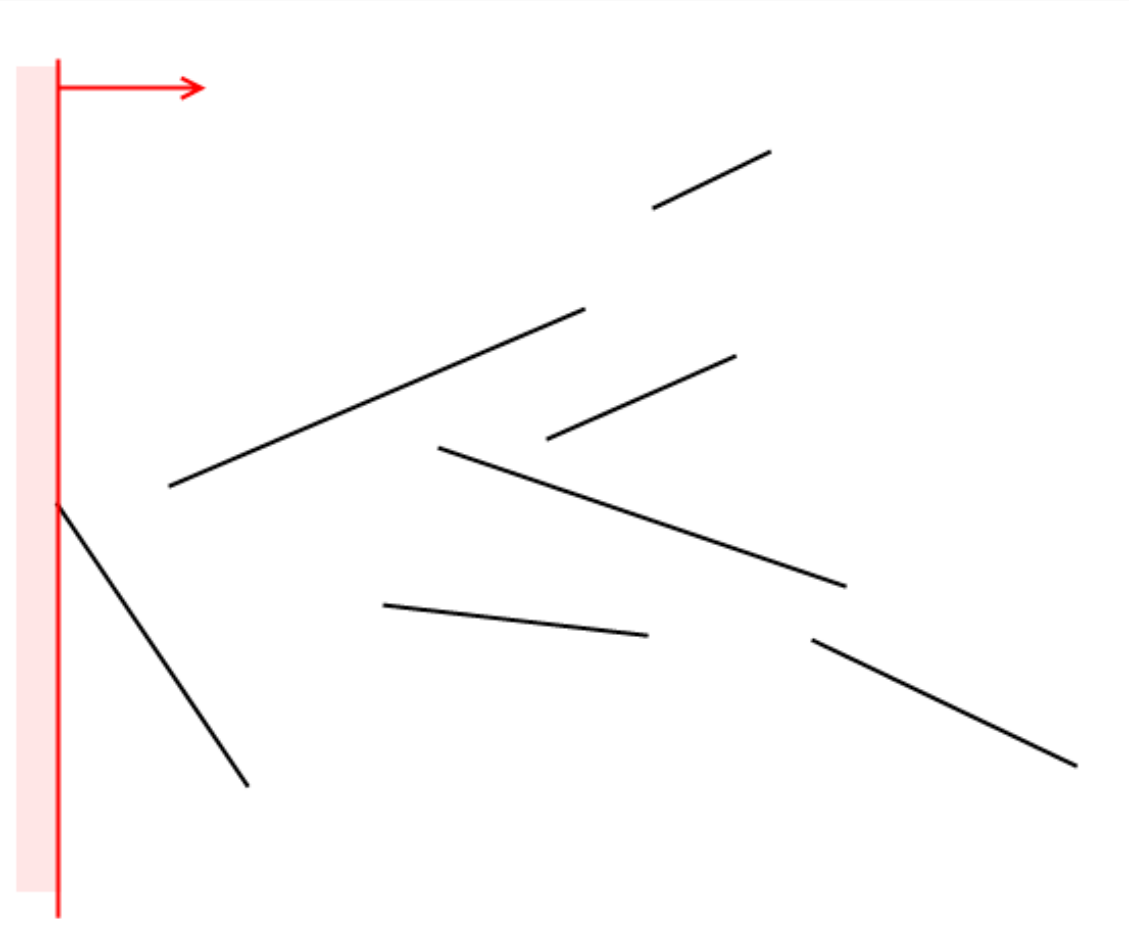
# Problem 5: Pinball

Where does the pinball end up? Up to $N \leq 100,000$ non-intersecting non-flat line segment obstacles.



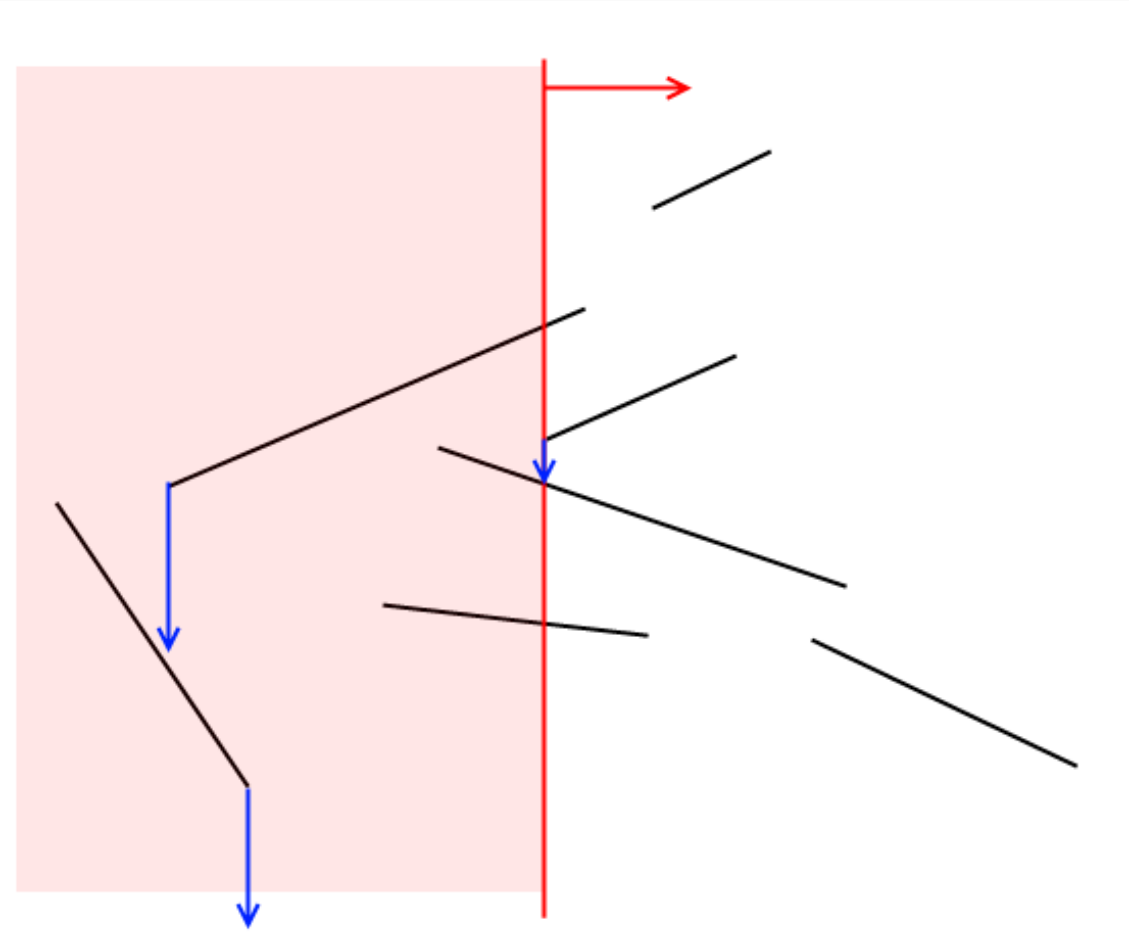
# Problem 5: Solution

Sweep left to right, find the “next segment” that a pinball on each segment would fall onto from the lower end point.



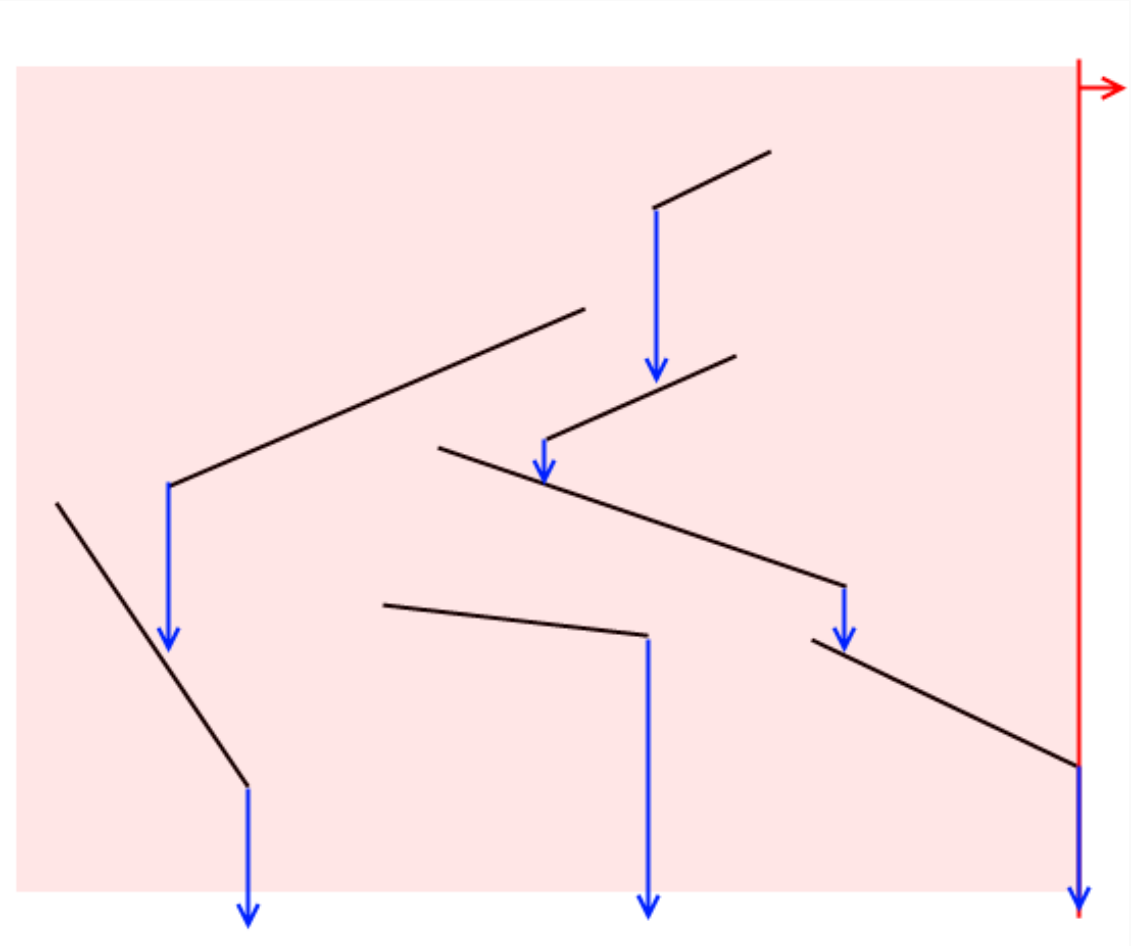
# Problem 5: Solution

Sweep left to right, find the “next segment” that a pinball on each segment would fall onto from the lower end point.



# Problem 5: Solution

Sweep left to right, find the “next segment” that a pinball on each segment would fall onto from the lower end point.



# One Sweep to Rule Them All

Exercise: solve these problems by line sweep

- Arbitrary polygon intersection and union
- Triangulate a non-convex polygon
- Cut a convex polygon by lines
- Find closest pair of points
- Find all regions divided by $k$ infinite lines
- Construct a Voronoi Diagram

(some have easier solutions; line sweep is complicated to code!)

# Next Class

More Line Sweep