Line Segment Intersection Using Sweep Line Algorithm

Saba Naji
Anahita Hedayatifard

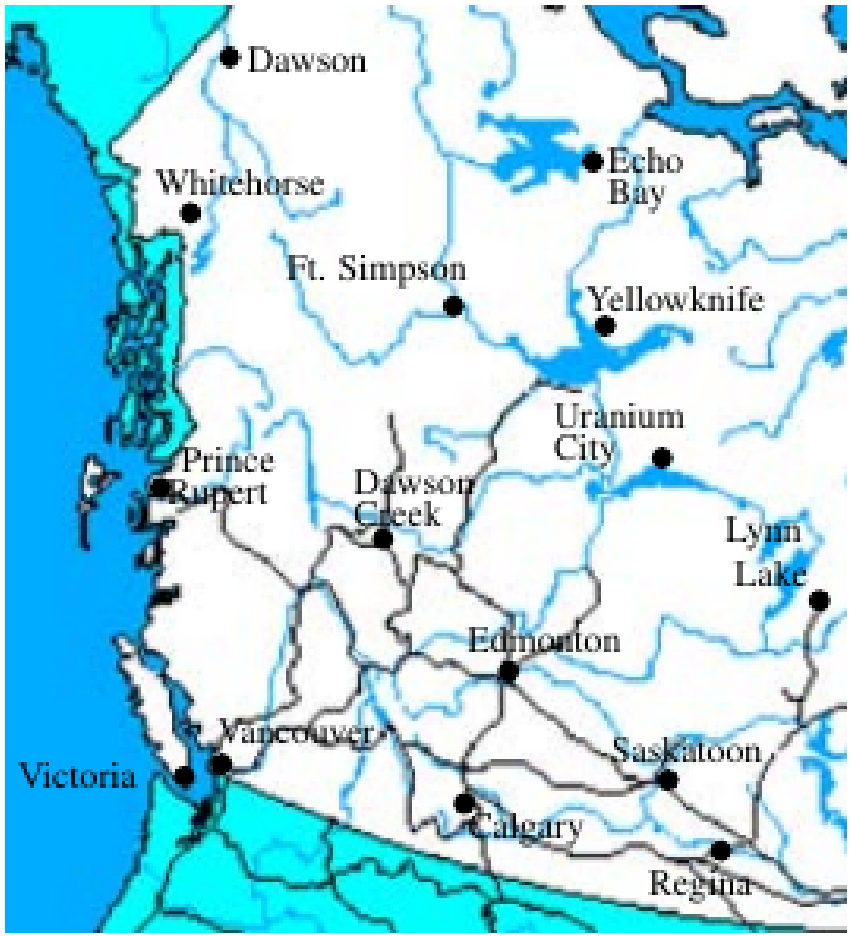
Professor: Dr. Shakournia
Department of Computer Science,
University Of Tehran

July, 2024

Table of Contents

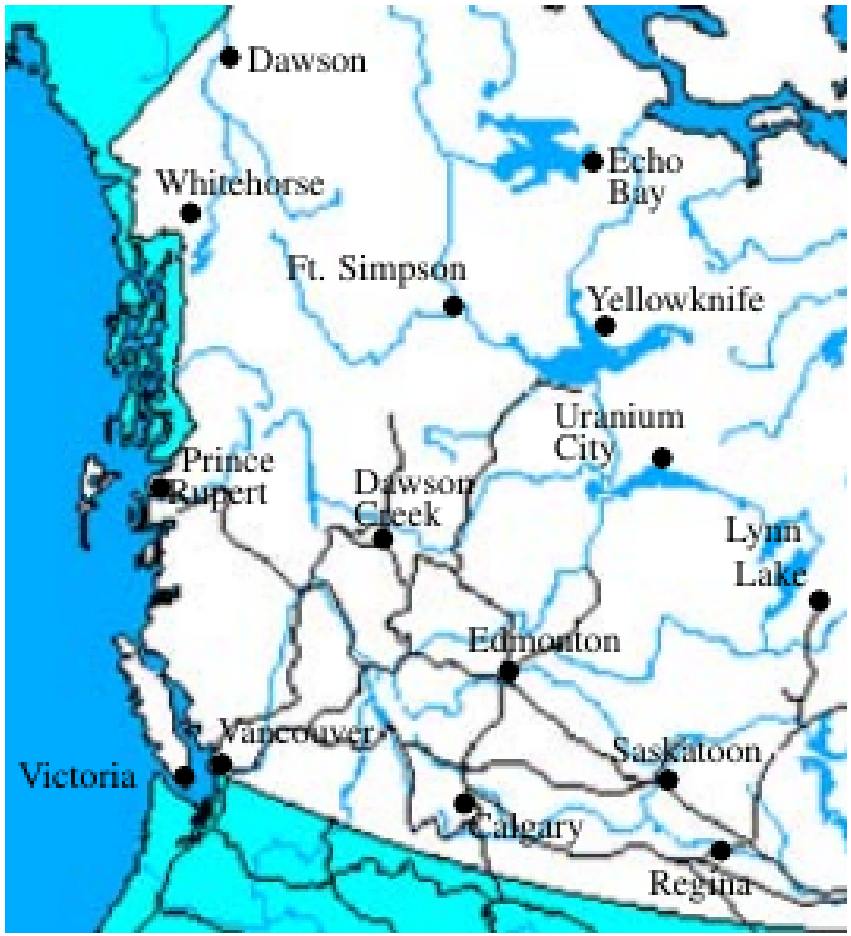
Thematic Map Overlay

- GIS allows users to create, manage, analyze, and map all types of data.



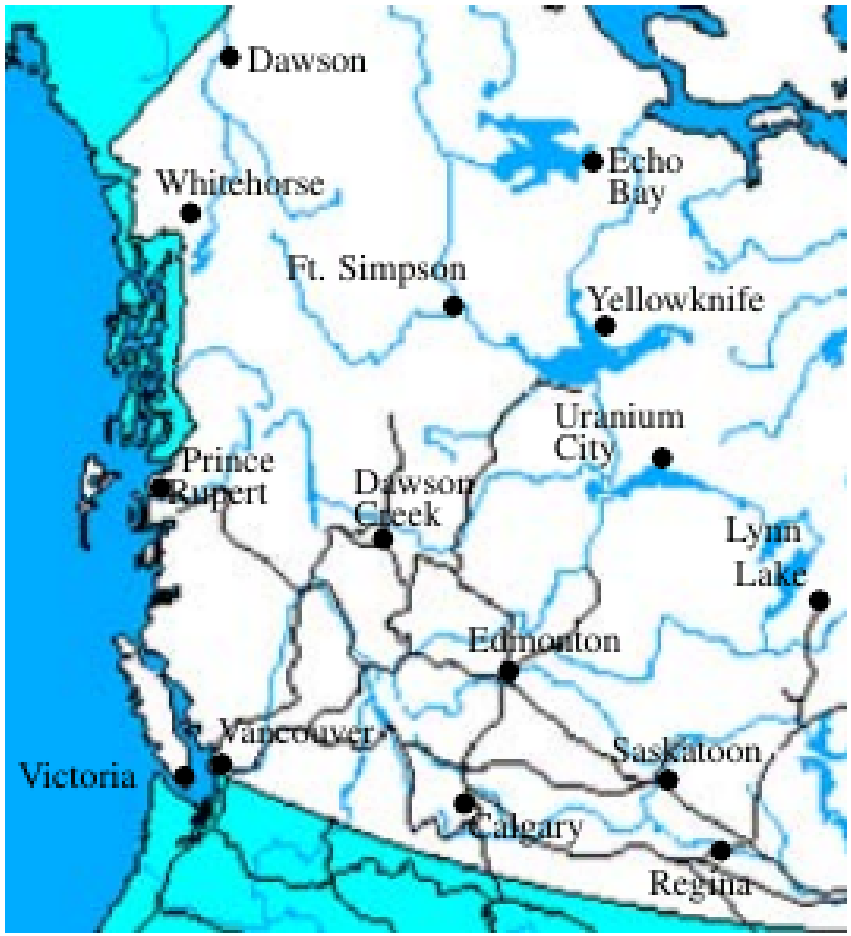
Thematic Map Overlay

- GIS allows users to create, manage, analyze, and map all types of data.
- Each type of information (e.g., roads, rivers) is stored in separate layers.



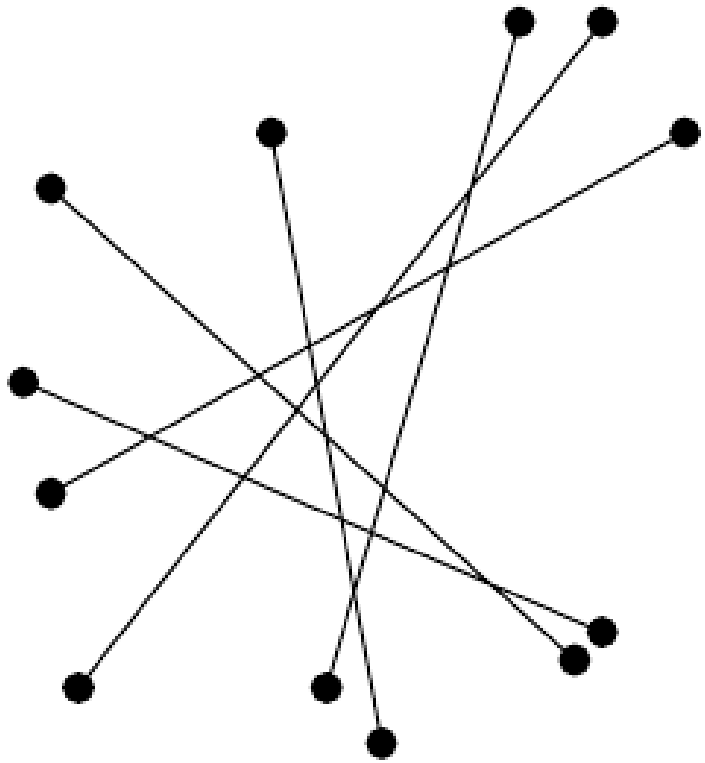
Thematic Map Overlay

- GIS allows users to create, manage, analyze, and map all types of data.
- Each type of information (e.g., roads, rivers) is stored in separate layers.
- Intersections are positions of special interest: For example, in the overlay of road and river layers.



Restatement of the Problem

- Given two sets of line segments, compute all intersections between a segment from one set and a segment from the other.



- **Input:** A set S of line segments in the plane.

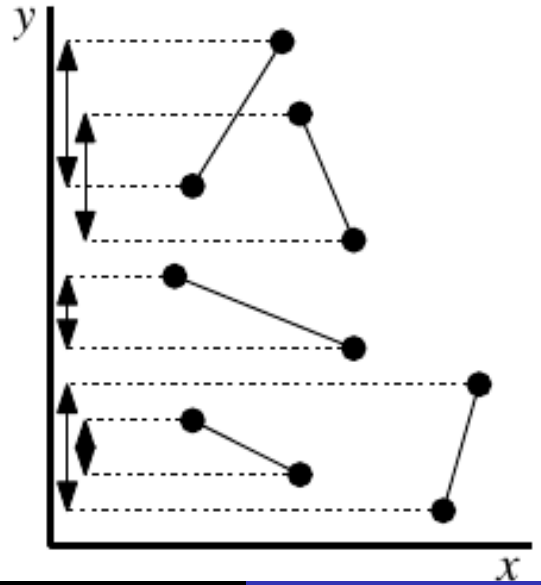
- **Input:** A set S of line segments in the plane.
- **Ouput:** The set of intersection points among the segments in S, with for each intersection point the segments that contain it

Ways of Solving the Line Segment Intersection Problem

- **Brute-force**: simply take each pair of segments, compute whether they intersect, and, if so, report their intersection point.($O(n^2)$)

Ways of Solving the Line Segment Intersection Problem

- **Brute-force**: simply take each pair of segments, compute whether they intersect, and, if so, report their intersection point.($O(n^2)$)
- **Sweep Line**: segments that are close together are candidates for intersection, unlike segments that are far apart. ($O((n+k)logn)$ where k is the size of the output)



Sweep Line Algorithm

- Imagine sweeping a horizontal line downwards over the plane.

Sweep Line Algorithm

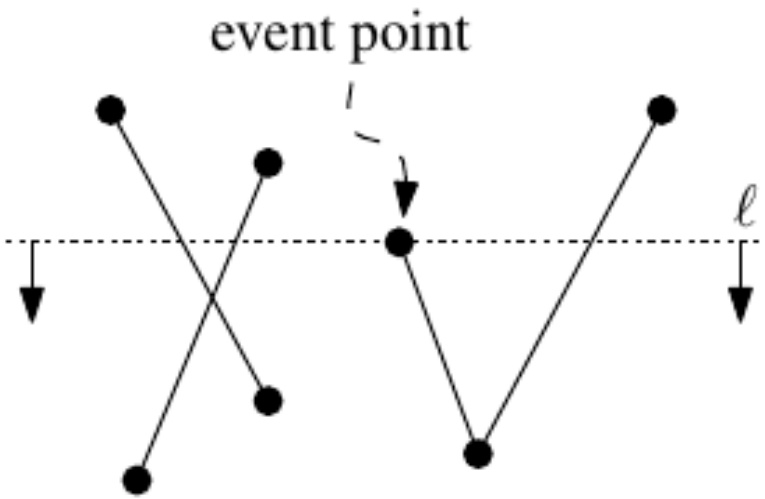
- Imagine sweeping a horizontal line downwards over the plane.
- Maintain an ordered sequence of segments intersecting the sweep line.

Sweep Line Algorithm

- Imagine sweeping a horizontal line downwards over the plane.
- Maintain an ordered sequence of segments intersecting the sweep line.
- Event points: Endpoints of segments and intersection points.

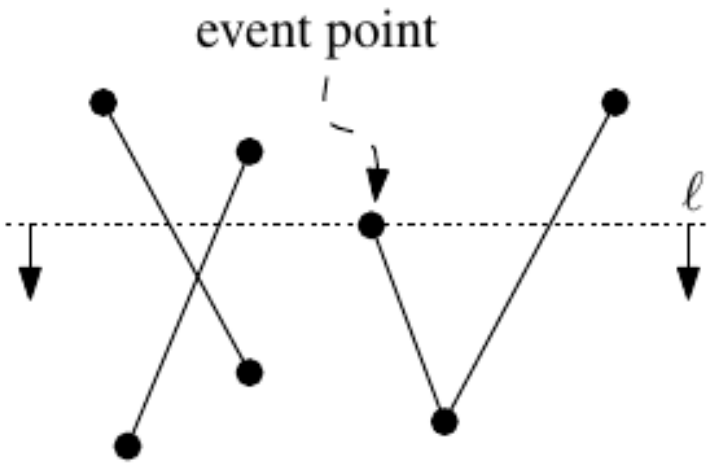
Sweep Line Algorithm

- Imagine sweeping a horizontal line downwards over the plane.
- Maintain an ordered sequence of segments intersecting the sweep line.
- Event points: Endpoints of segments and intersection points.



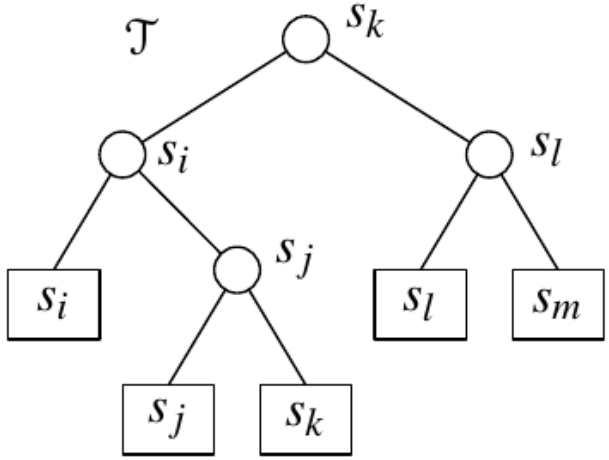
- After sweeping the entire plane and treating the last event point, all intersection points are computed.

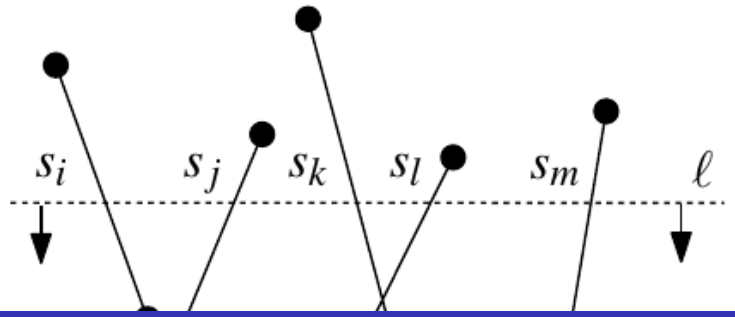
- After sweeping the entire plane and treating the last event point, all intersection points are computed.
- Invariant: All intersection points above the sweep line are correctly computed at any given time during the plane sweep.



Data Structures

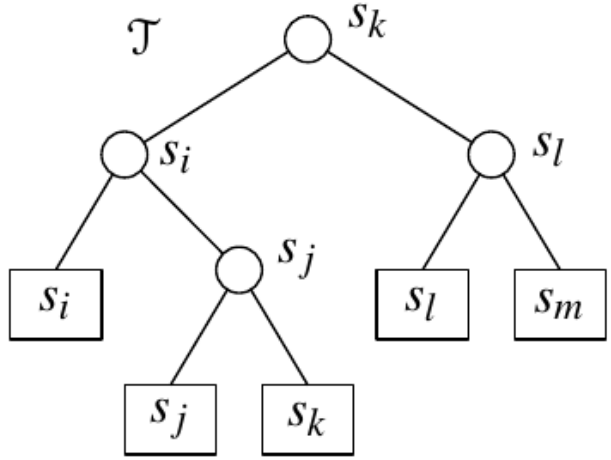
- **Event Queue (Q):**
  - Stores upcoming events (endpoints and intersections).
  - Sorted by y-coordinate.

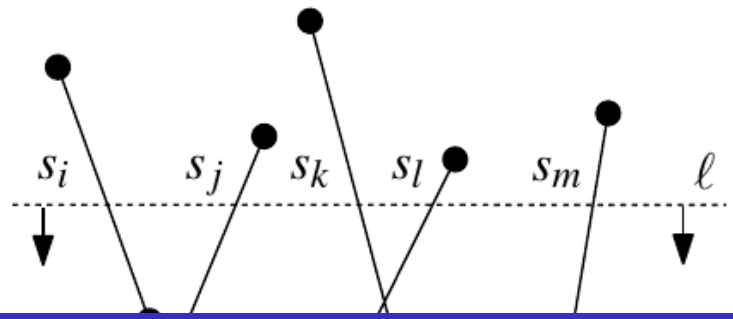




Data Structures

- **Event Queue (Q):**
  - Stores upcoming events (endpoints and intersections).
  - Sorted by y-coordinate.
- **Status Structure (T):**
  - Maintains current intersecting segments.
  - Sorted by x-coordinate at the sweep line.





Event Points in Plane Sweep

- **Upper Endpoint:**
  - New segment intersects the sweep line.
  - Test for intersections with adjacent segments.

Event Points in Plane Sweep

- **Upper Endpoint:**
  - New segment intersects the sweep line.
  - Test for intersections with adjacent segments.
- **Lower Endpoint:**
  - Segment stops intersecting the sweep line
  - Test new adjacent segments for intersections.

Event Points in Plane Sweep

- **Upper Endpoint:**
  - New segment intersects the sweep line.
  - Test for intersections with adjacent segments.
- **Lower Endpoint:**
  - Segment stops intersecting the sweep line
  - Test new adjacent segments for intersections.
- **Intersection Point:**
  - Two segments swap positions.
  - Test new adjacent segments for intersections.

Handling Event Points

- **Upper Endpoint:**
  - Add new segment to status.
  - Test for intersection with neighbors.

Handling Event Points

- **Upper Endpoint:**
  - Add new segment to status.
  - Test for intersection with neighbors.
- **Lower Endpoint:**
  - Remove segment from status.
  - Test new neighbors for intersection.

Handling Event Points

- **Upper Endpoint:**
  - Add new segment to status.
  - Test for intersection with neighbors.
- **Lower Endpoint:**
  - Remove segment from status.
  - Test new neighbors for intersection.
- **Intersection Point**
  - Swap segment order.
  - Test new neighbors for intersections.

Algorithm Steps

- **Initialization:**
  - Insert all segment endpoints into event queue Q.
  - Initialize empty status structure T.

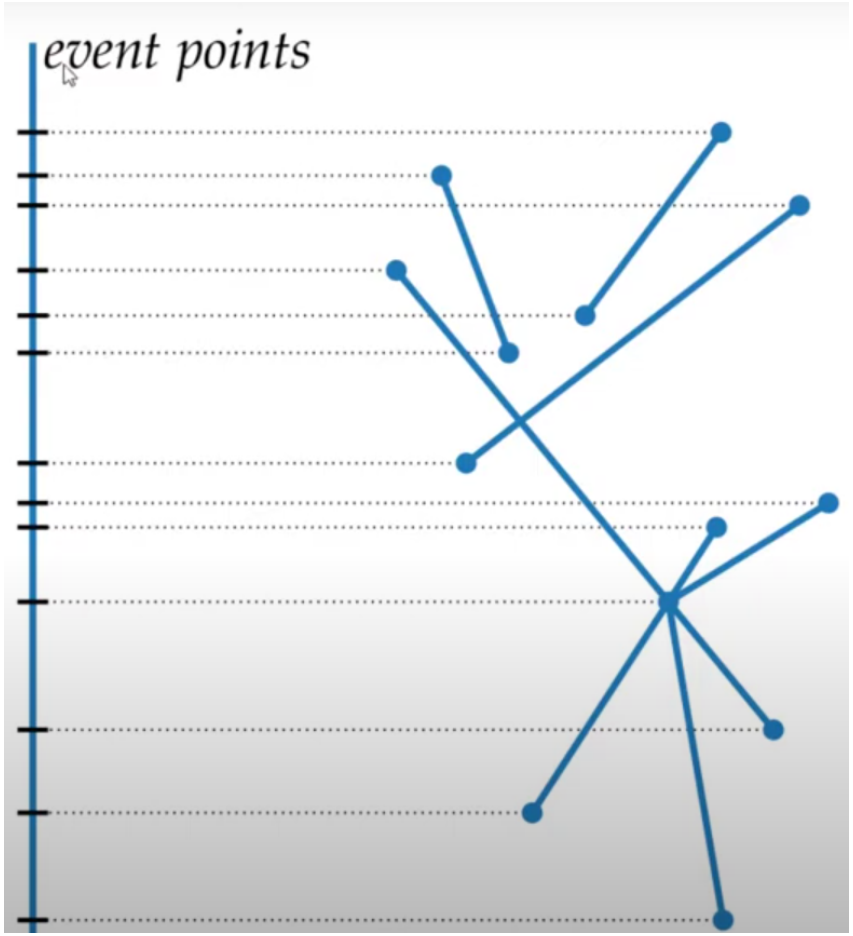
Algorithm Steps

- **Initialization:**
  - Insert all segment endpoints into event queue Q.
  - Initialize empty status structure T.
- **Processing:** While Q is not empty
  - Extract the next event point p.
  - Handle event at p.

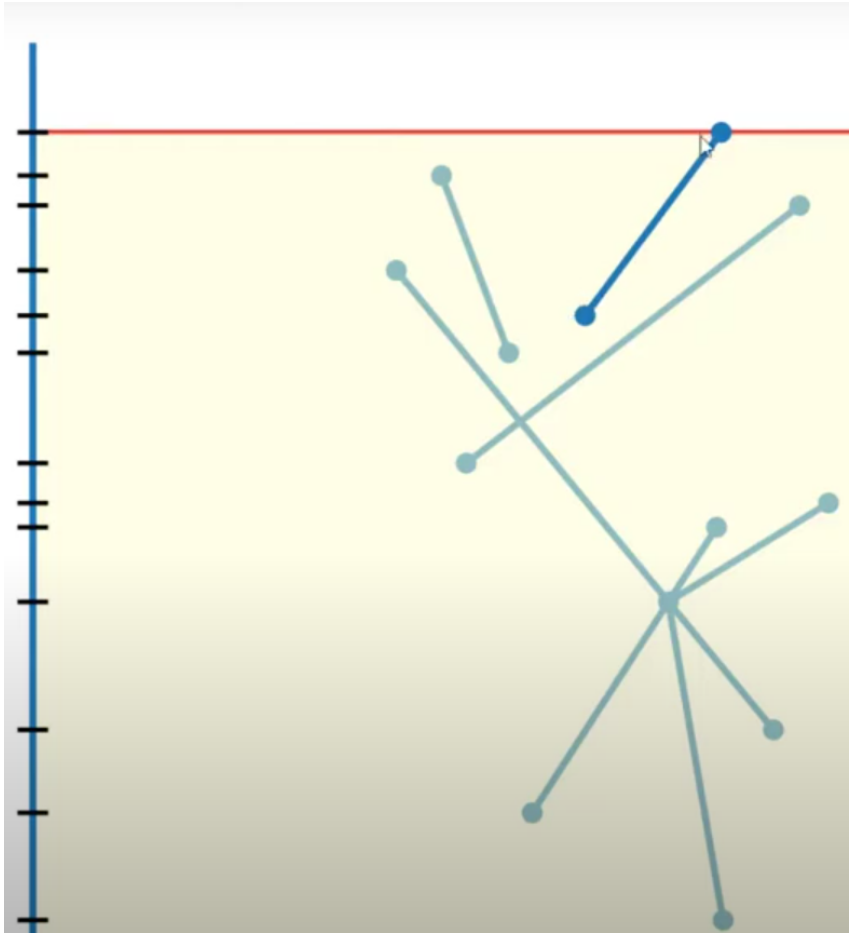
Algorithm Steps

- **Initialization:**
  - Insert all segment endpoints into event queue Q.
  - Initialize empty status structure T.
- **Processing:** While Q is not empty
  - Extract the next event point p.
  - Handle event at p.
- **Handling Events:**
  - Update status structure T.
  - Test for new intersections.

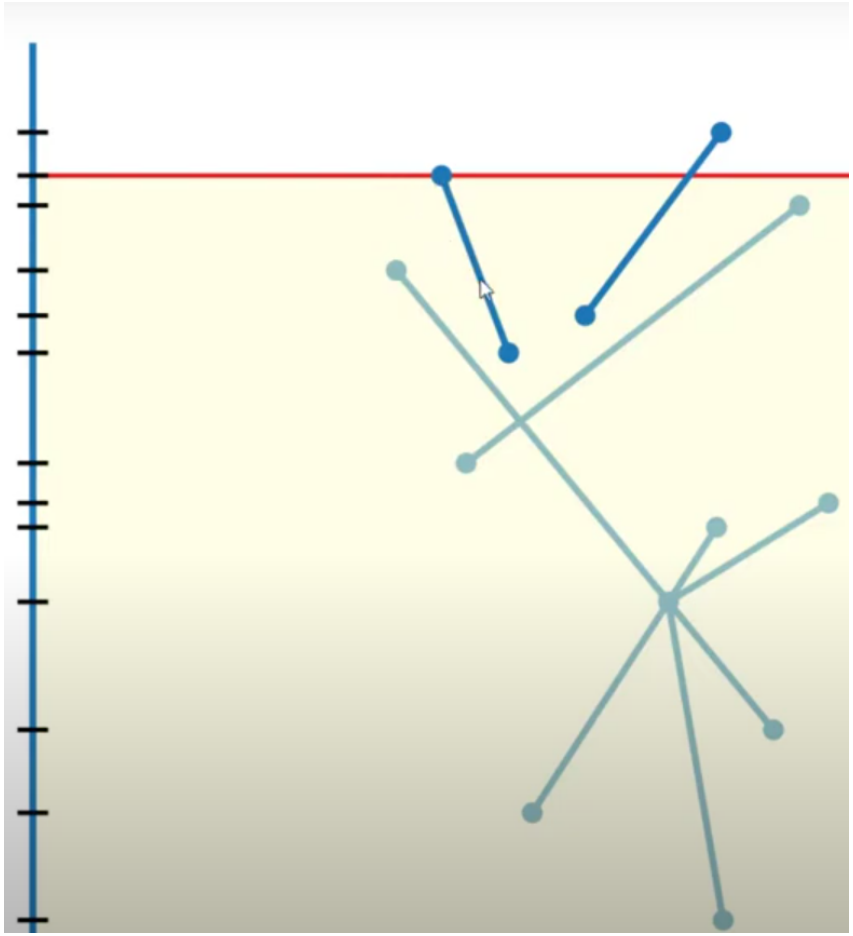
An Example



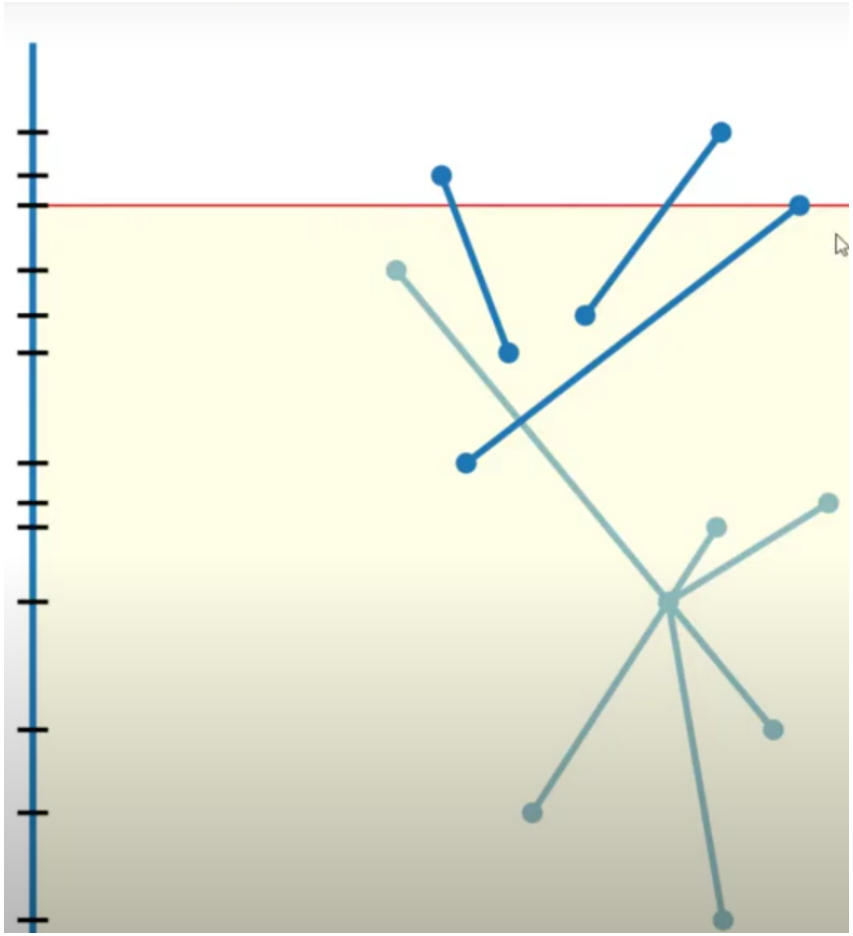
An Example



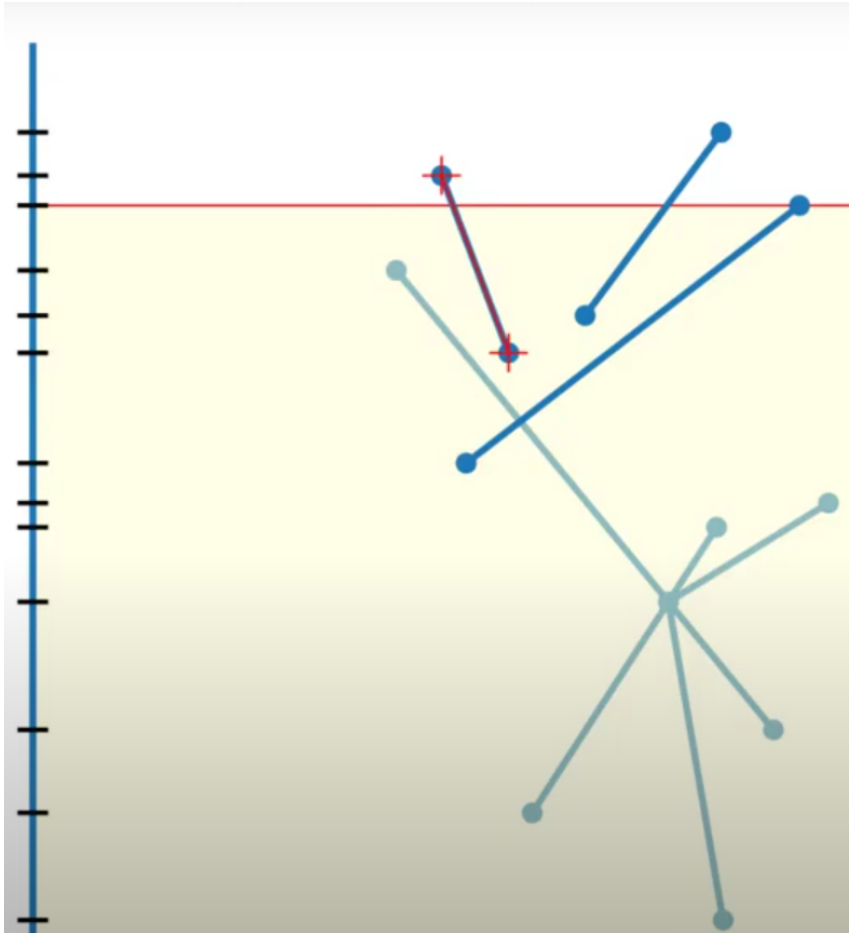
An Example



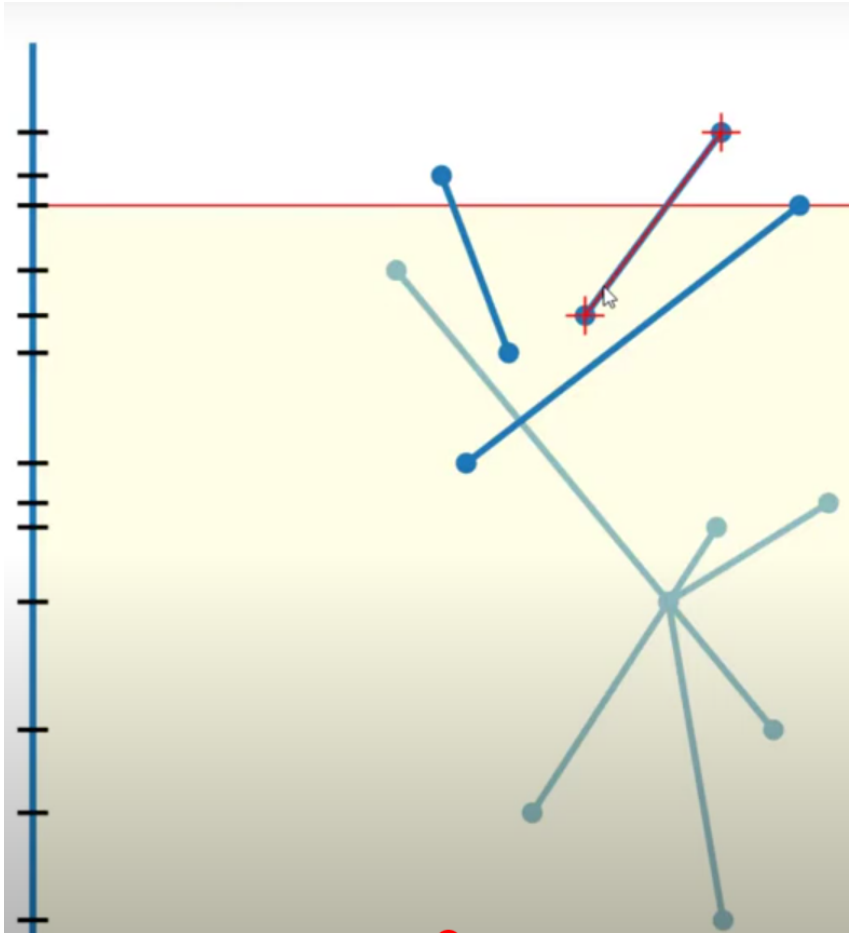
An Example



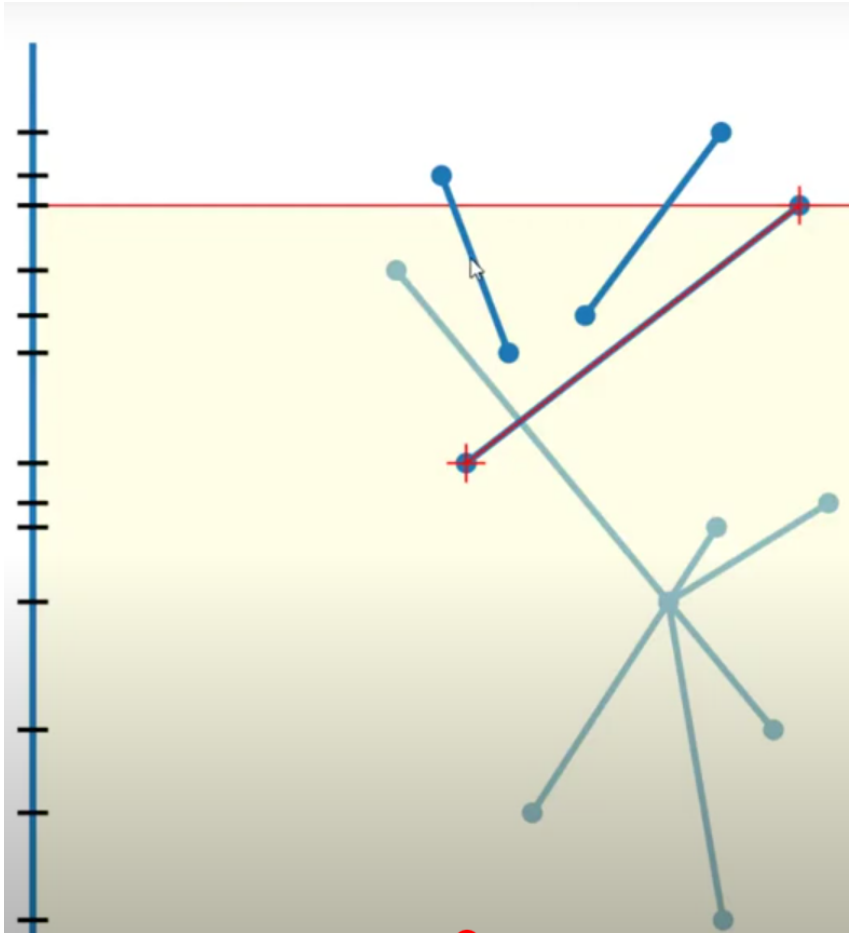
An Example



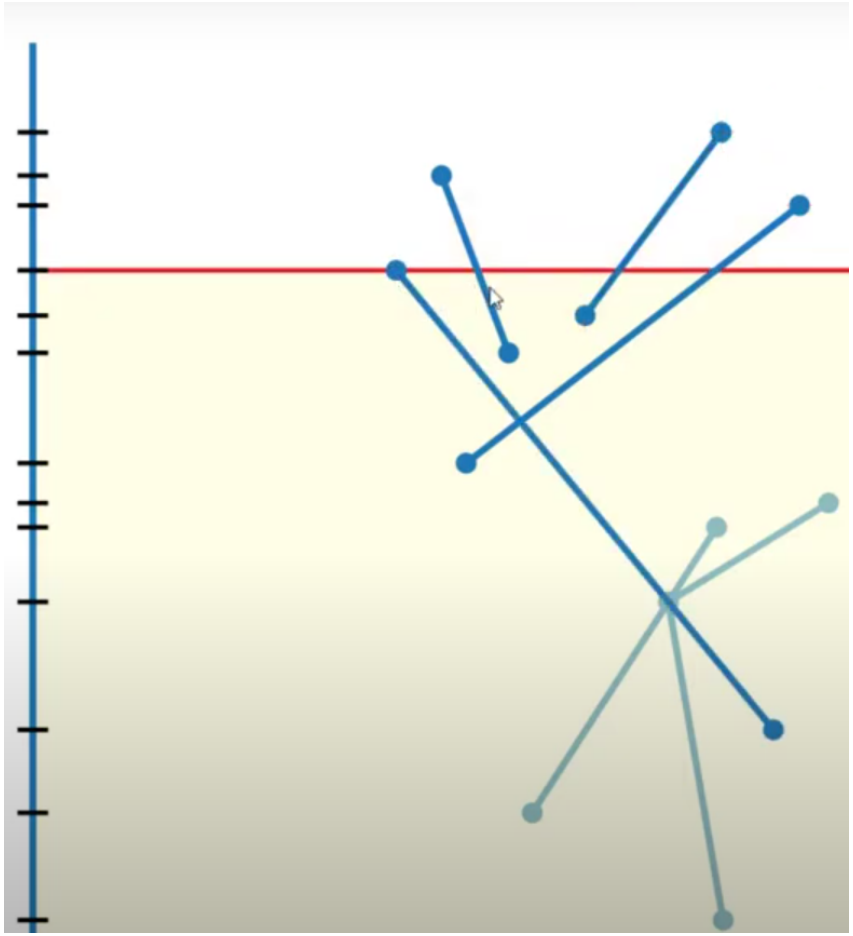
An Example



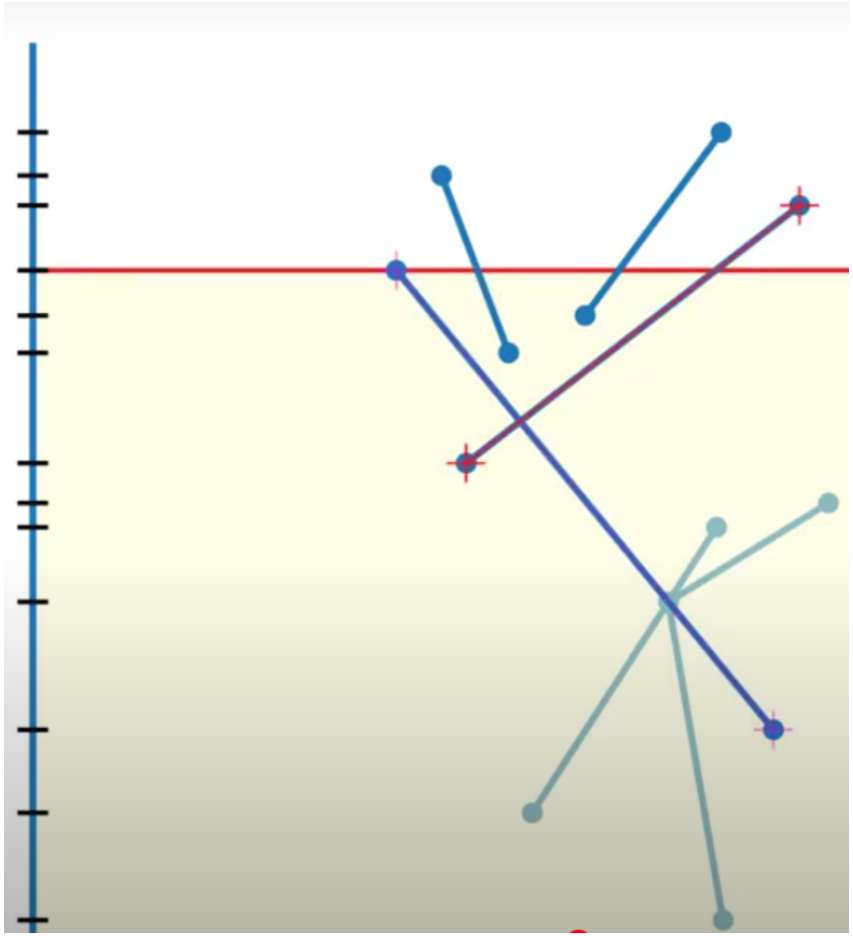
An Example



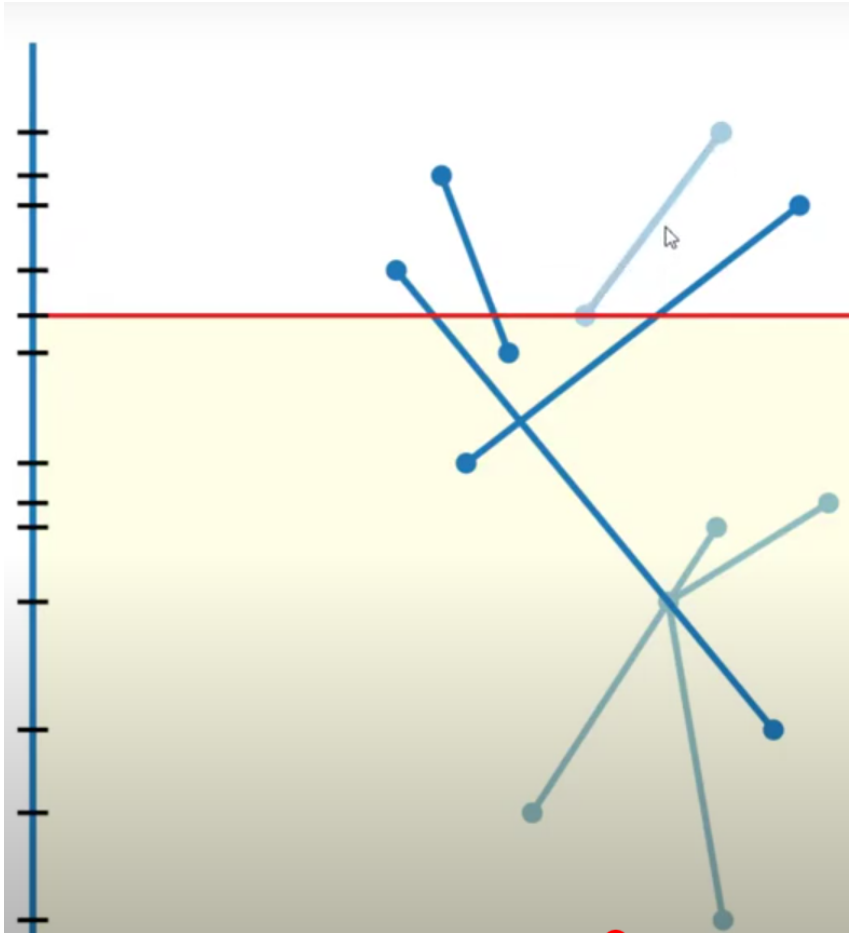
An Example



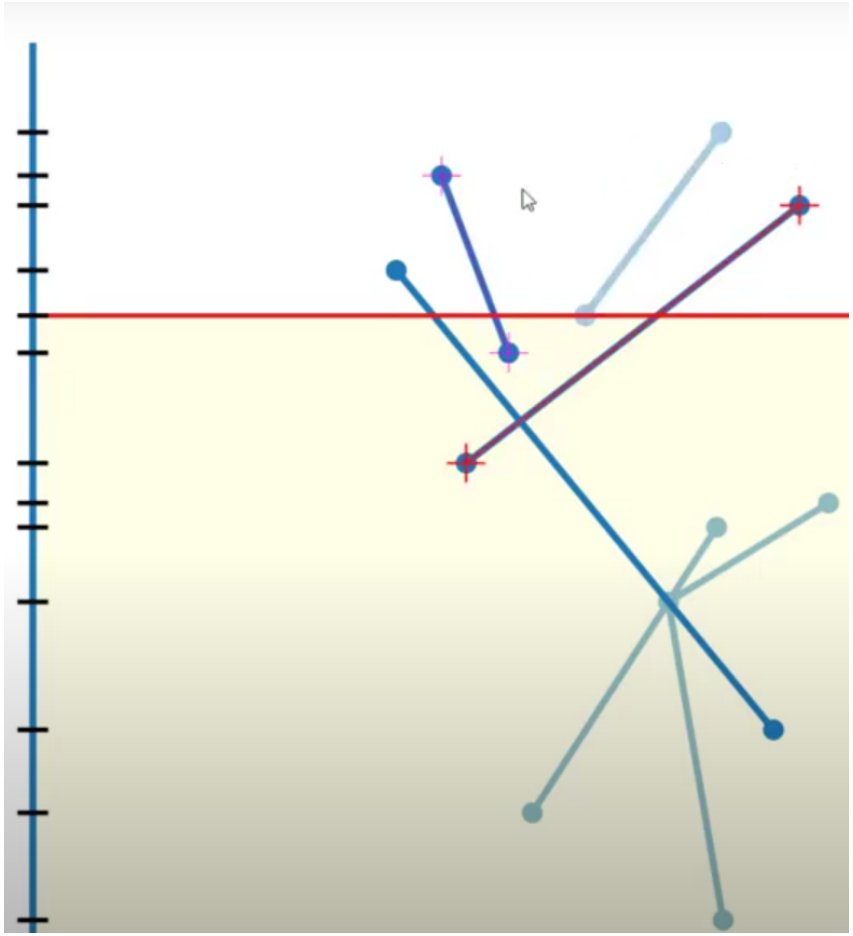
An Example



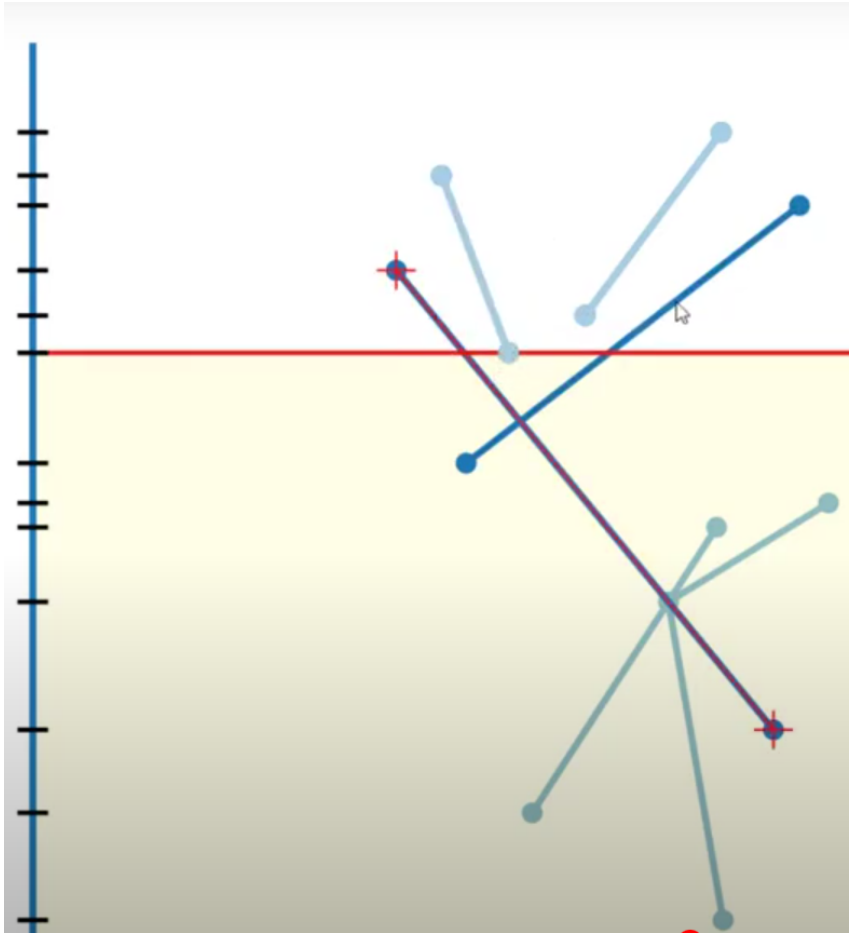
An Example



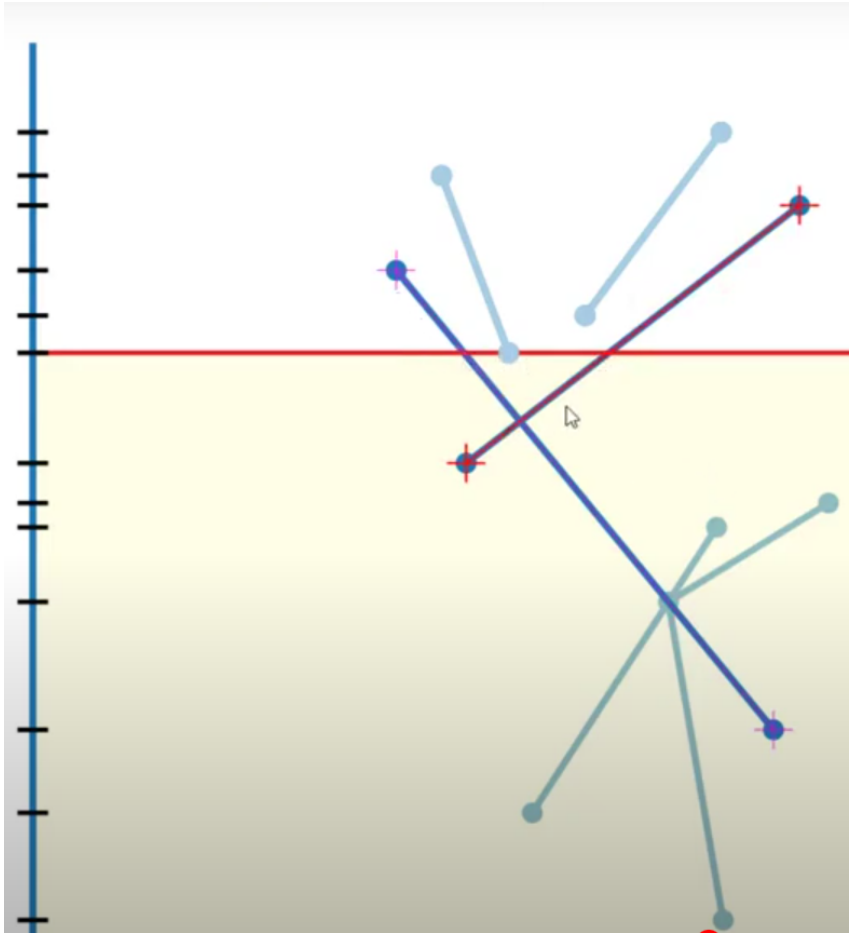
An Example



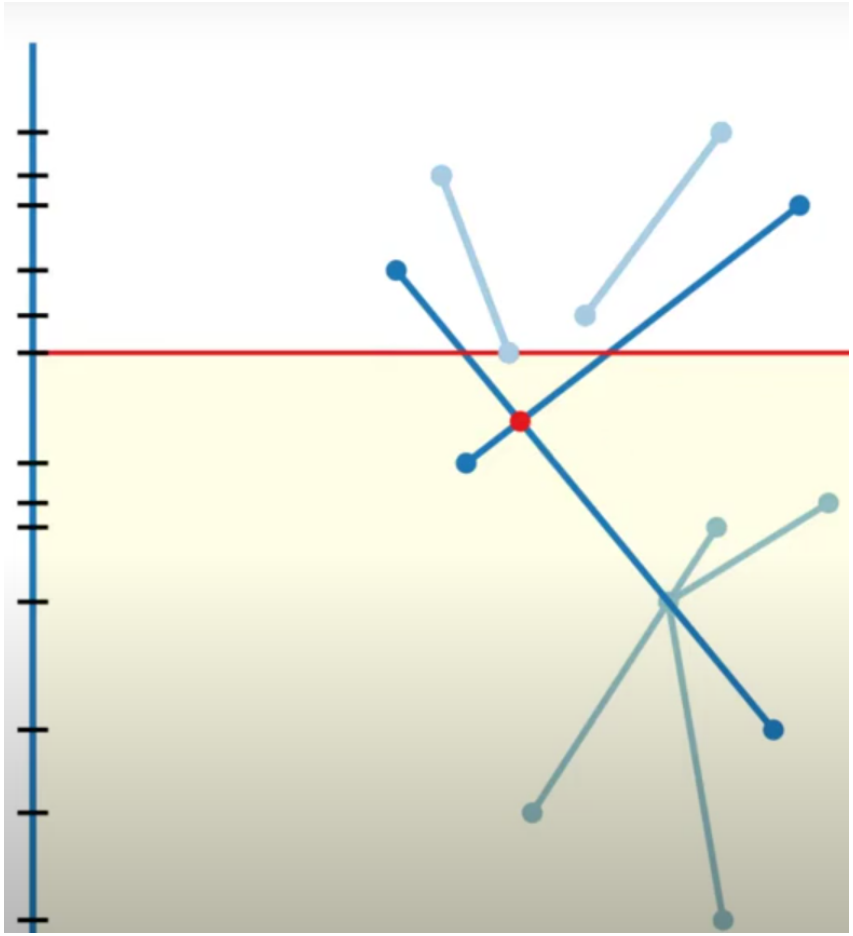
An Example



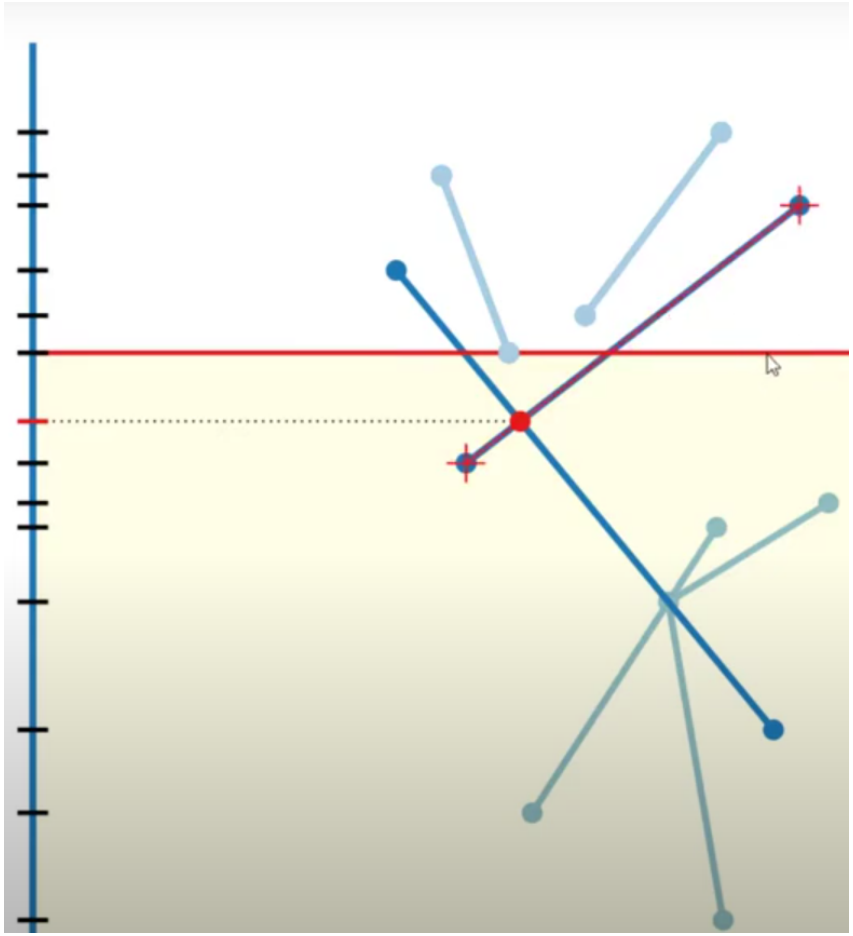
An Example



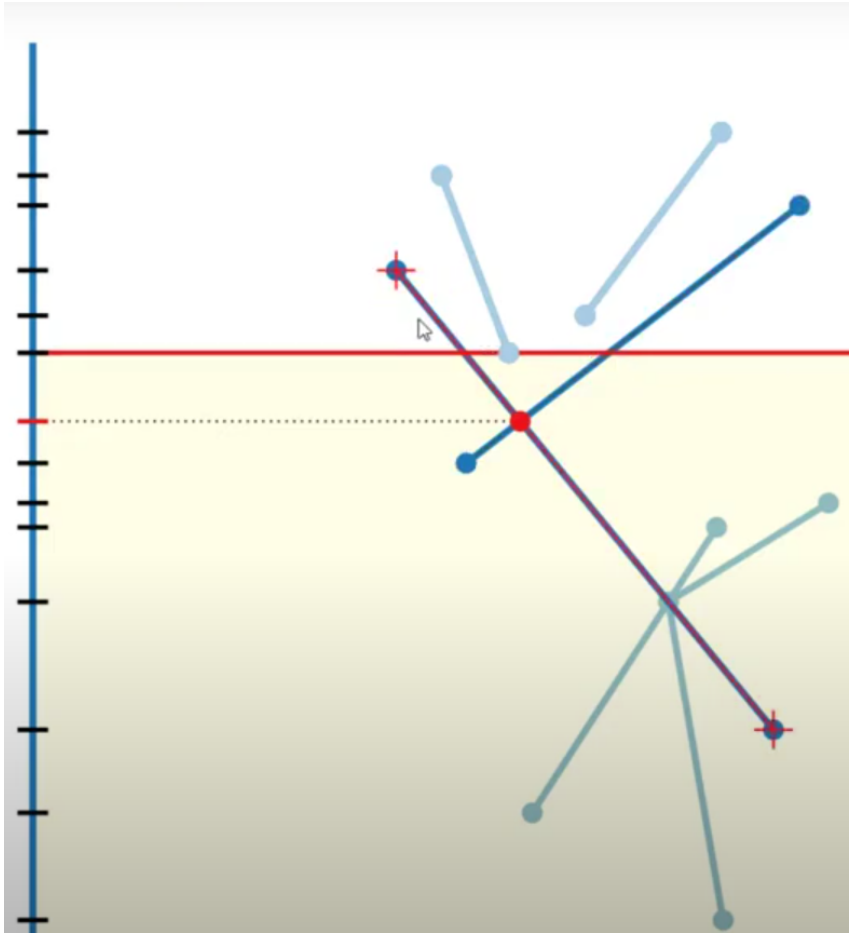
An Example



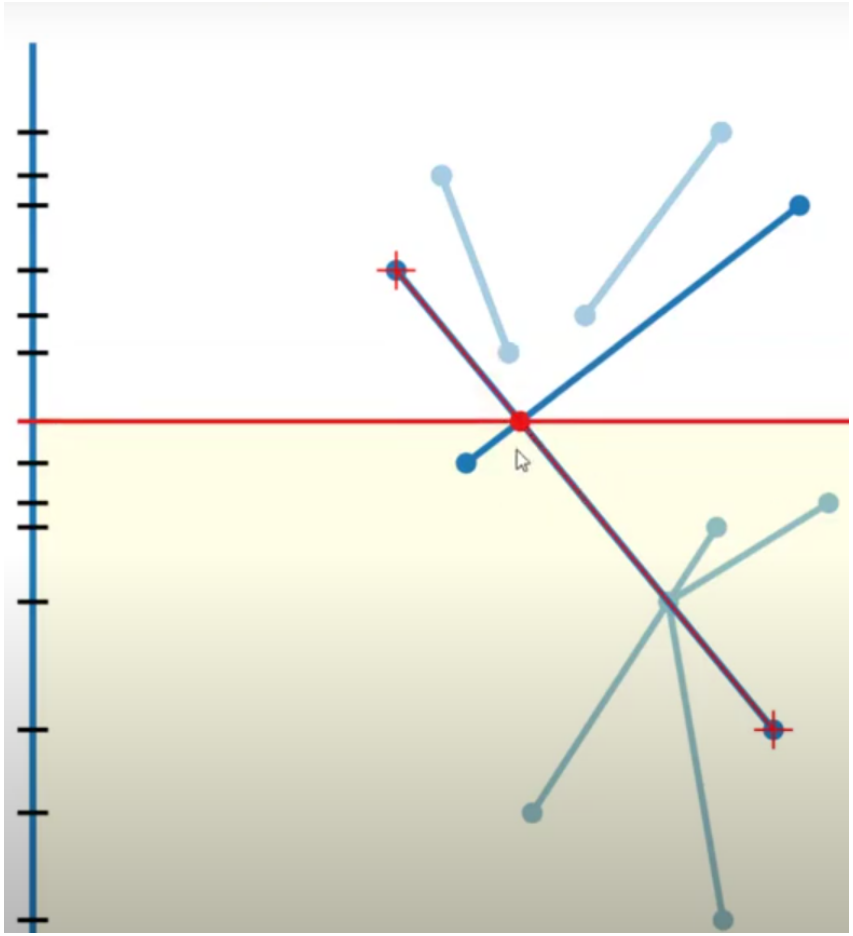
An Example



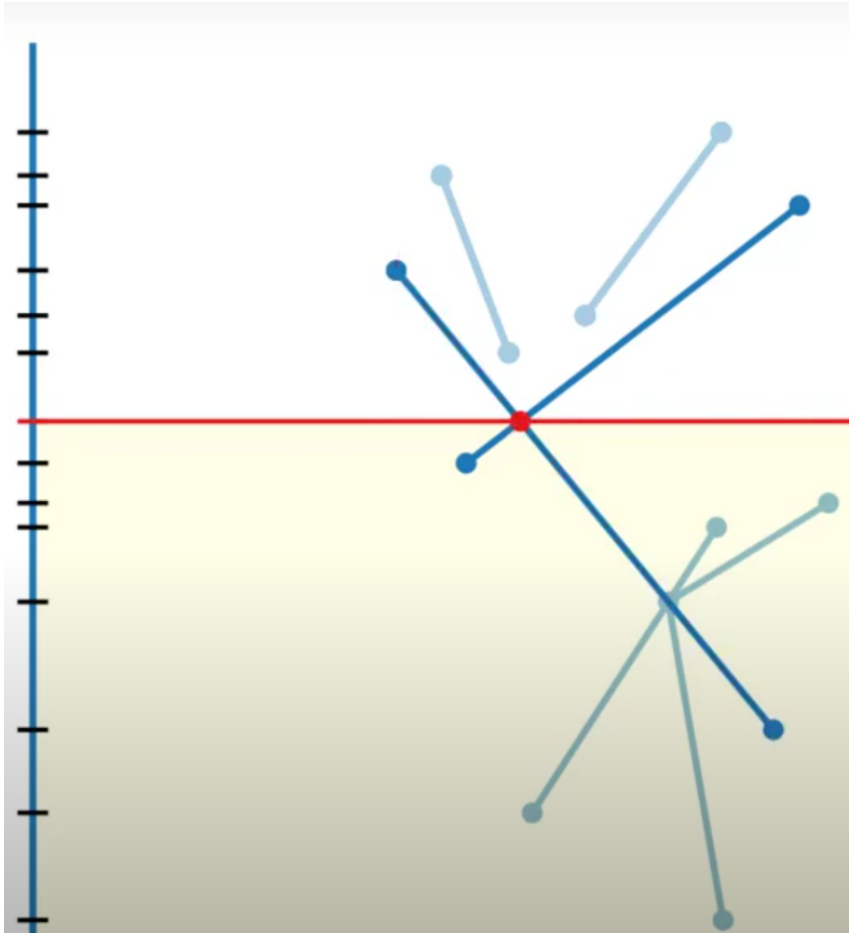
An Example



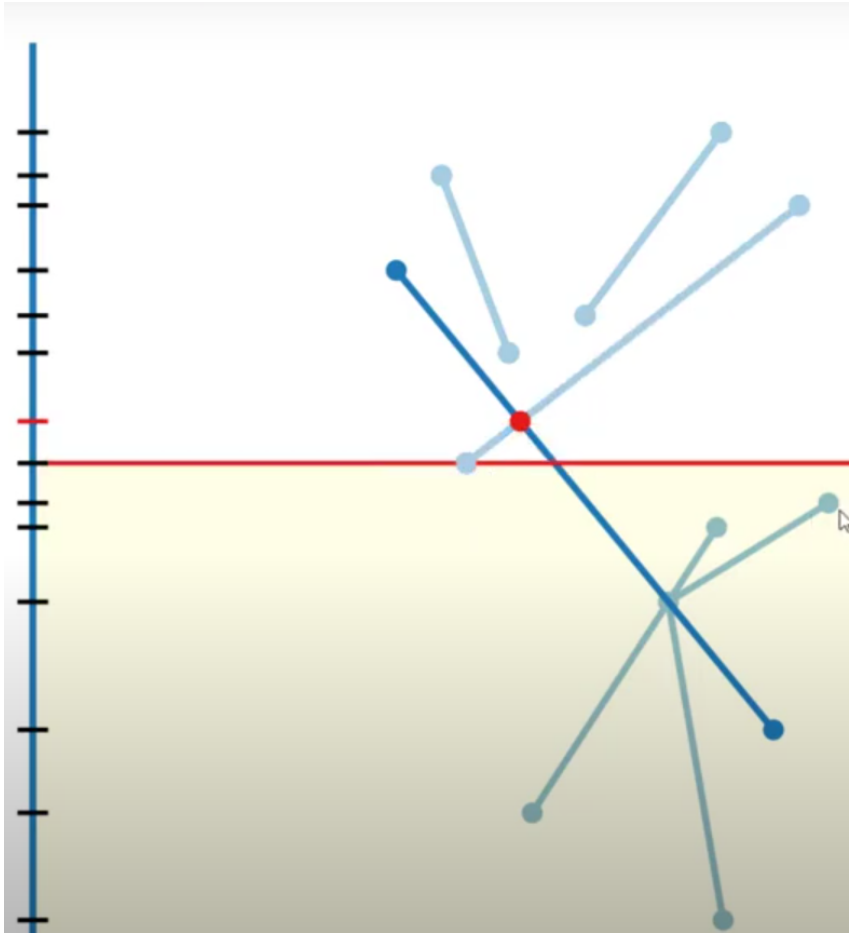
An Example



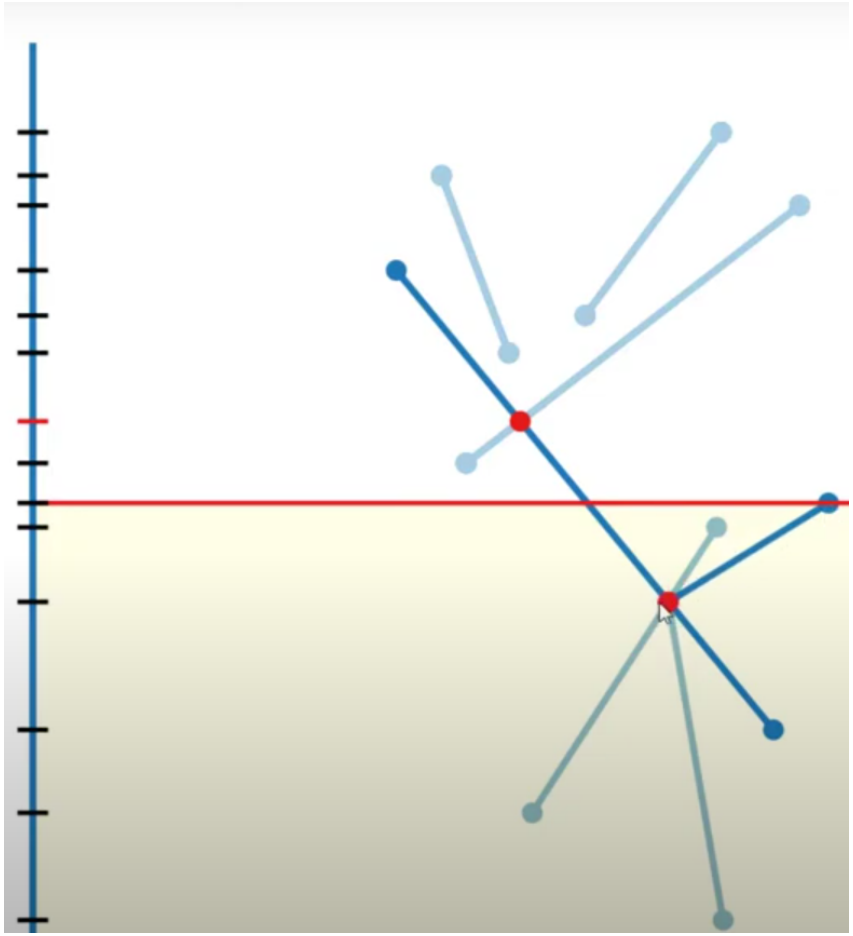
An Example



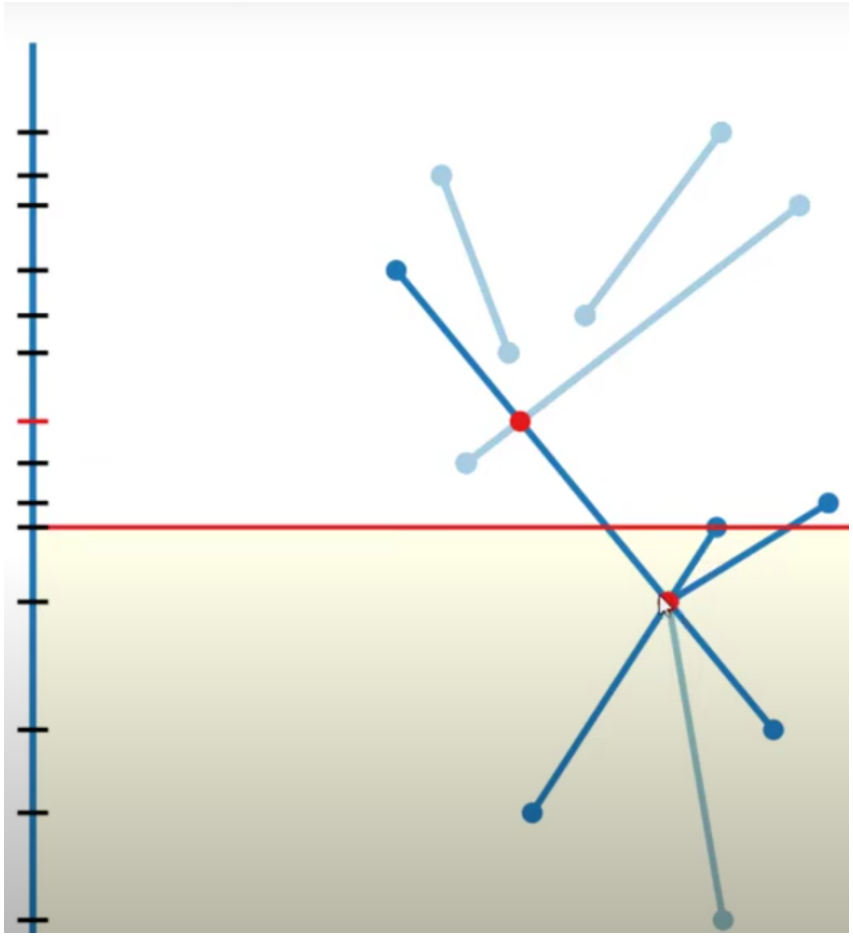
An Example



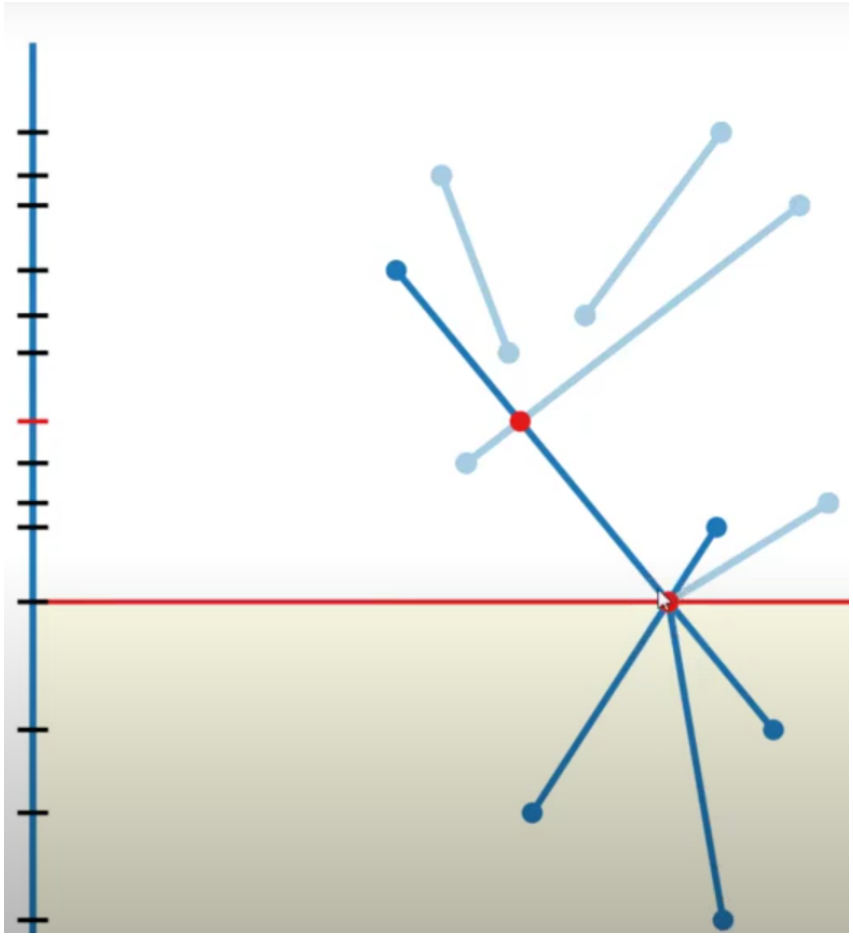
An Example



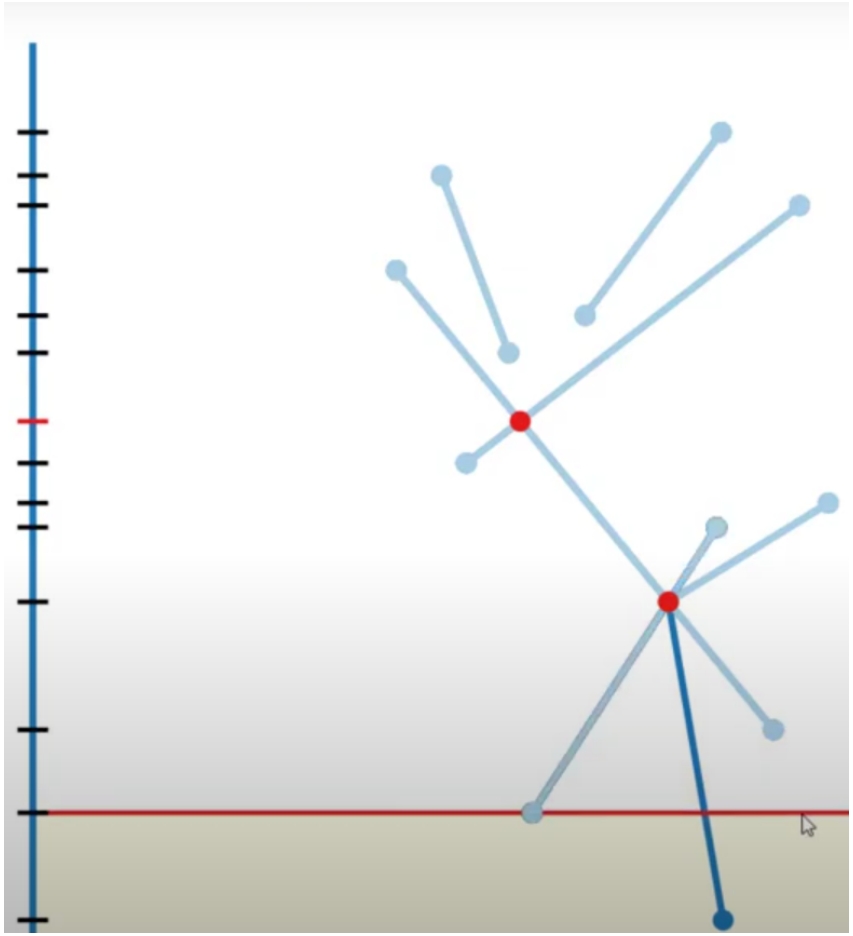
An Example



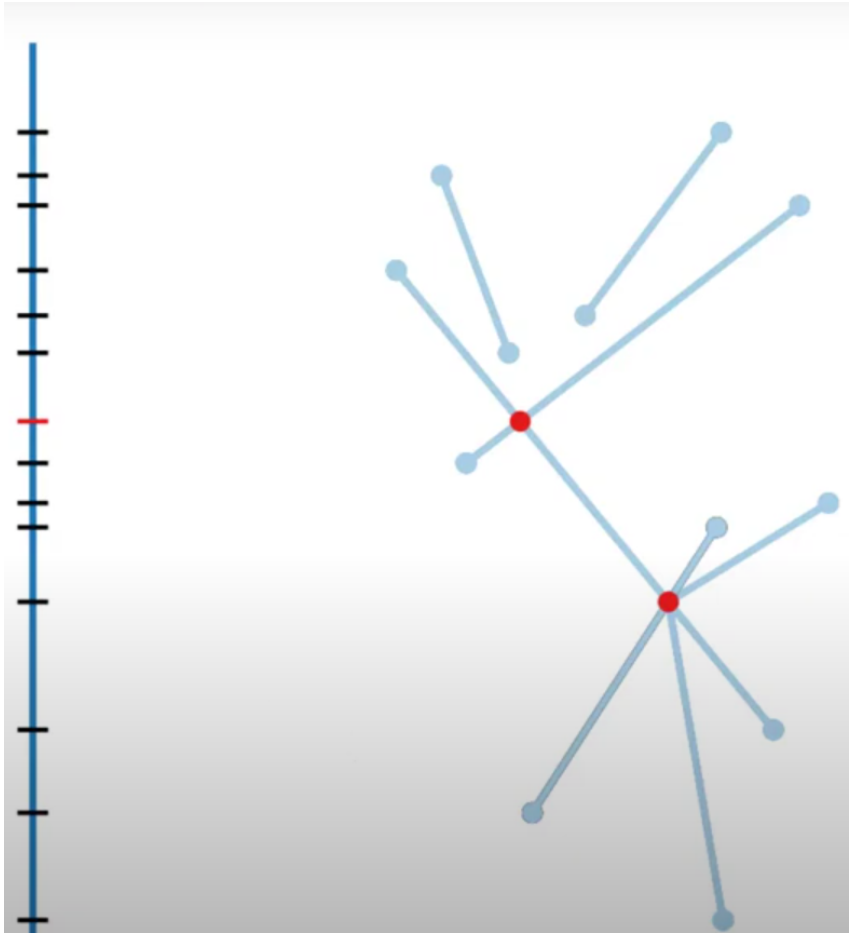
An Example



An Example



An Example



References


de Berg, Mark and Cheong, Otfried and van Kreveld, Marc and Overmars, Mark, *Computational Geometry: Algorithms and Applications*, Springer 978-3-540-77973-5, 2008.


Bartuschka, Ulrike and Mehlhorn, Kurt and Nn, Stefan, *A Robust and Efficient Implementation of a Sweep Line Algorithm for the Straight Line Segment Intersection Problem*, 1997


Sweep-Line Algorithm for Line Segment Intersection


Plane Sweep Algorithm for finding Line Segment Intersections