Assignment 2:
Computational Geometry.

## Line Sweep Algorithm.

**Classes:**

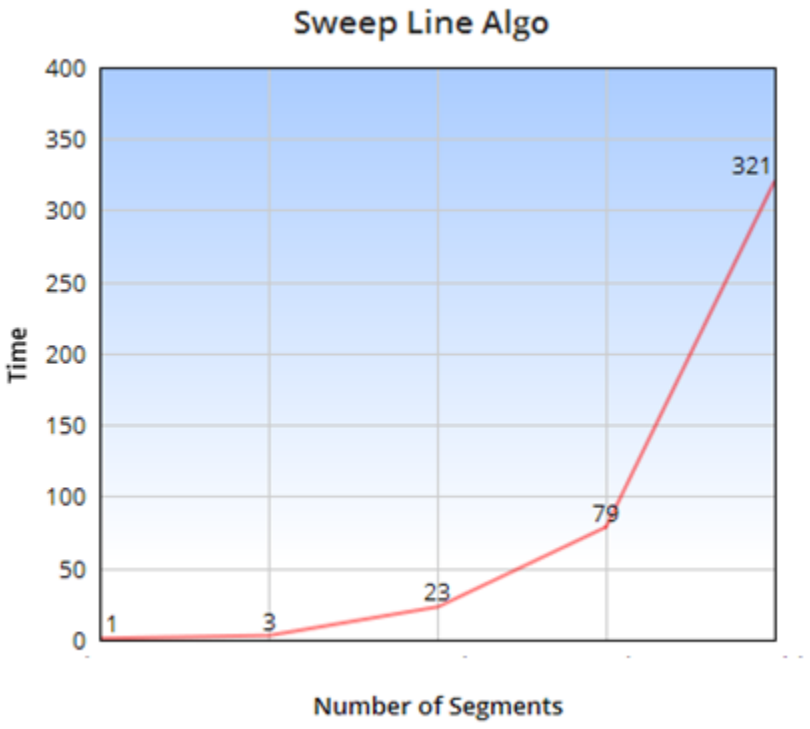
1. PlaneSweep: main class.
2. Point: (double x, double y) representing a point.
3. Segment: point a, point b representing a segment.
4. Event: String (type of event), segment a, segment b, point (event point)
   Type of events: Start of segment, end of segment, intersection of two segments.
5. EventQ: TreeSet that stores Events ordered with increasing x.
6. SweepLineStatus: Tree set for sweepline status.

**Data Structures:**

1. EventQ and
2. SweeplineStatus.
   Using T**reeSet** data structures for the above two. This is important as they are needed to be ordered with respect to increasing x.

**Run Time:**

The runtime depends primarily on the eventQ, which is a TreeSet and thus any function performed on it takes O(Log(n)) time.
For the algorithm we take O((2n+I)logn) where I is the number of intersections.
This happens n times because of n events in the EventQ.



Assignment 2:
Computational Geometry.

## Brute Force:

**Classes:**

1. BruteForce: main class.
2. Point: (double x, double y) representing a point.
3. Segment: point a, point b representing a segment.

**Data Structures:**

1. HashSet: stores the calculated intersection points between segments.

**Run Time:**

We check each segment's equation with other segment's equation.
Thus we check $n^2$ times. Giving us a $O(n^2)$