**PROJECT PHASE II**

**By**

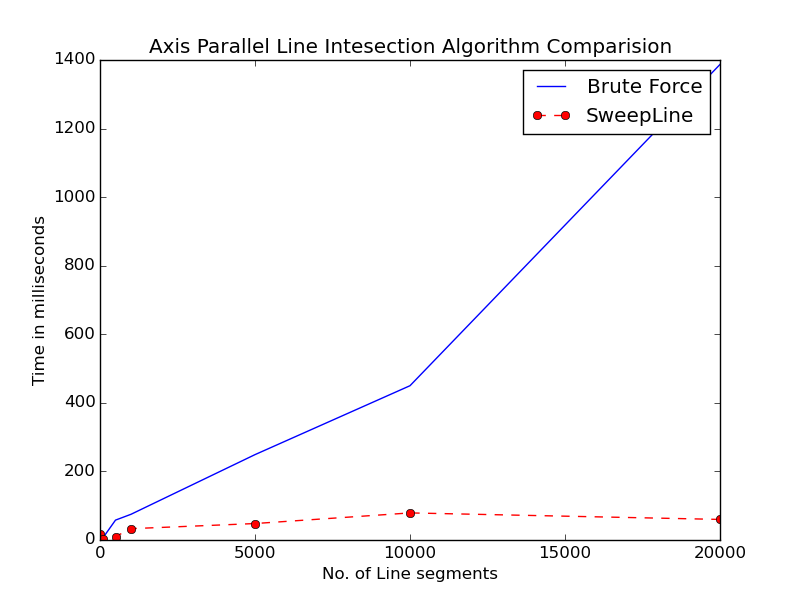
**Krati Nayyar**

**Siddharth Bhal**

**Reading the code:**

* **MainClass**: This is the main file for the code which is to be executed for different number of horizontal and vertical lines.
* **RandomLineGenerator**: This is the file which implements the code for random generation of Horizontal and Vertical Lines, using the classes **HorizontalLines** and **VerticalLines.**
* **BruteForce**: This class contains the code for executing the brute force method for finding the intersection of the lines.
* **HVIntersection**: This is the class which is executed for determining the horizontal and vertical lines intersection using the Sweep Line method.
* **RangeSearch :** It is used to determine the range using the Self Balancing Binary Search tree.
* **SegmentHV:** This is used to represent Horizontal and Vertical Line Segments, I.e., functions to find intersections of the line segments.

**Graph for Brute Force and Sweep Line Algorithm**

****

**Files attached with this document**

BruteForce.txt – Intersections found by Brute Force algorithm for all possible values of number of lines.

SweepLine.txt – Intersections found by sweep line algorithm for all possible values of lines. The intersection found by brute force and sweep line algorithm are the same.

RandomLines.txt – All the lines generated by random line generator class.

**Conclusion**

The time taken by Brute Force is greater than Sweep line algorithm as complexity of Brute Force is O(n2) whereas for SweepLine it is O(n log n + k)

As its visible that time taken by algorithm isn’t increasing uniformly with sample size. This is likely due to time recorded representing application time as well as system time. We only needed application spent time but couldn’t find a API to obtain it.