

# Code Explanation

Umesh Karthikeya

March 9, 2024

*helper.cpp* runs in  $O(n \log n)$  time using modified divide and conquer algorithm. I am storing line segments in a vector called *posters*. And each line segment has two points and the slope of the line segments. Then I am sorting the line segments using *comparePosters* help function in increasing order of starting x-coordinate (a) this step takes  $O(n \log n)$  time. *length* function is used to calculate the x-axis covered distance in  $O(n)$  time. Then *merger\_sort* function uses *mymerge* function to calculate the outline of the combined trapeziums. *merger\_sort* divide the posters into half recursively. *mymerge* will combine them.

In *mymerge* function have two arguments *left* and *right* which will merged these are two outlines formed by few trapeziums. Here there are around 26 cases which I have figured out on my own. First I am classifying them based on x-axis co-ordinates. Then in each of the cases there are two more classifications whether line segments intersecting or not. And in each of the cases there are three classifications based on height. Covering all cases of merging as far as I figured out. This merge step uses two iterators for each vector and do merge operation in  $O(n)$  time. Therefore our *merger\_sort* calculation function has,

$$T(n) = 2T(n/2) + O(n)$$

It's time complexity is also  $O(n \log n)$

Overall time complexity  $O(n \log n) + O(n) + O(n \log n)$  which is  $O(n \log n)$