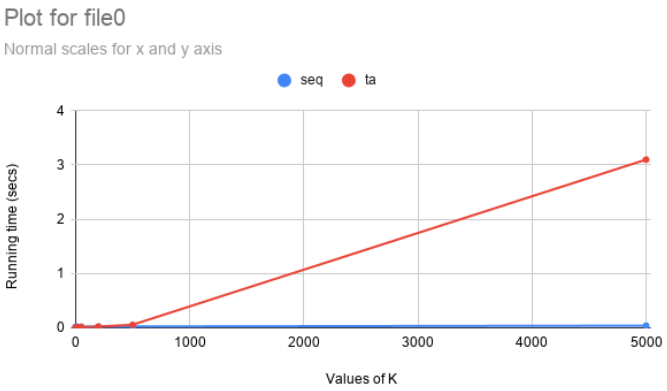
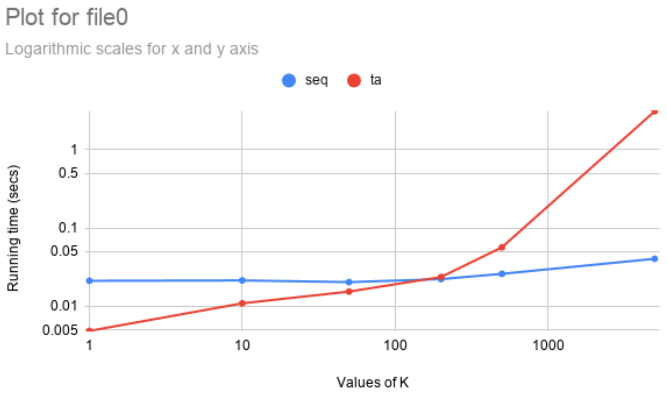
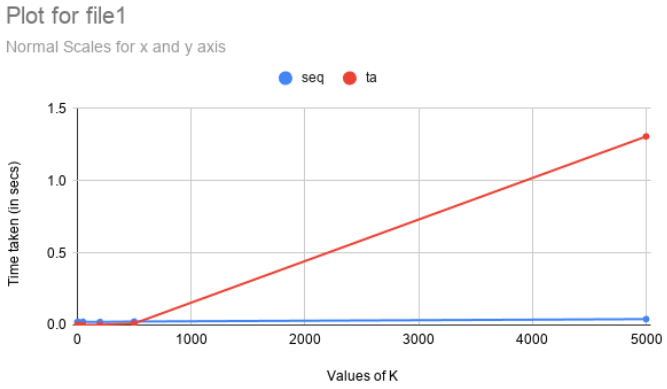
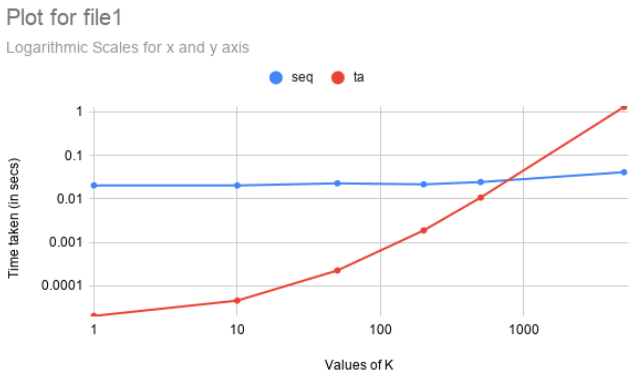
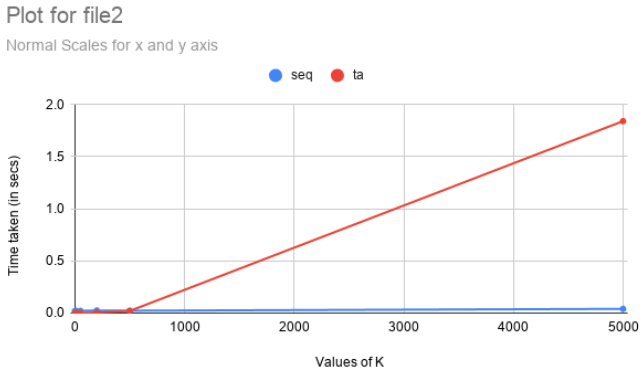
- The plots obtained in the case of file0, file1, and file2 are as follows:

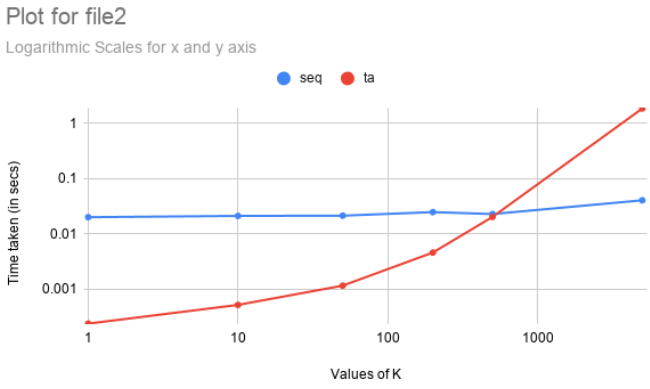












- Here you can find these different images corresponding to a normal scale of these values as well as according to the logarithmic scale for these values. These plots were computed using google sheets (link provided in the README)
- Here we can see that the time taken for the threshold algorithm is initially lesser than the time taken for the sequential algorithm for lower values of K. But for higher values of K, the time taken for the threshold algorithm just increases exponentially, while the time taken for the sequential algorithm almost remains a constant.
- The reason for the sequential algorithm to have a nearly constant time here is that it is meant to go through the entire dataset once and will only be required to maintain a minheap of size K. Thus the running time for sequential algorithm will be: O(nlog(k)). And thus, because of having a fixed n here; not much differences happen in the running time of the algorithm since the values of k are in general much lesser than the size of the database.
- If we look at the plots or the values of the running times included in the google sheets, then we can see that the running time of sequential algorithm is not affected much. And this is

because it has to go through the entire dataset once and so the main factor determining its run time will simply be the size of the data.

- But in the case of the threshold algorithm, the value of K gets really important here as we can see from the plots. From the plots or the results stored, we can also see that the file on which the data is run also determines the running time of the threshold algorithm.
- It was found that in the case of file1, the performance of the thresold algorithm was the best. The performance for this algorithm was slightly lower in the case of file2. And it was much worse than both these two in the case of file0. These trends can even be seen from the different plots included.
- The main reason for these differences in the threshold algorithm is that it is actually dependant on the data to a large extent. After sorting the data with respect to both the individual coordinates, we go through the maximal tuples which are obtained corresponding to these sorted tuples. And this algorithm terminates when the maximum bound obtained from this maximum tuple is smaller than the smallest of the top-k values obtained.
- From this stopping criterion, we can see that based on the data distribution; it can happen that the algorithm comes across this criterion very early and then it will immediately terminate. Say for example, we are given a data distribution with one outlier and wish to compute the maximum. Now after the individual sorting; the algorithm will take this outlier value initially and then will discover that there is no scope of having any element bigger than this and so it will terminate immediately. The same arguments will follow for the other larger values of K.
- Thus if the data is very uniform and clustered then the performance of the threshold algorithm would be worser in comparison to the case where the data will be having a definite trend (say these data points will be having a cluster which would proably look like a scattered y=x plot) or maybe contains some heavy outliers, outweighing all the other points. And this is because of the stopping criterion of the threshold algorithm.
- And so because of this reason, the threshold algorithm is more performant on file1 than file0.
- Also we can see that the running time of the threshold algorithm seems to be exponential with respect to K, and it becomes much slower than the sequential algorithm for larger values of K. This is because of the stopping criterion which is dependant on K and the correspondingly increasing minheap operations which would take time dependant on the values of K.
- We also need to keep a check over the possible repetitions as well while storing these pairs to the minheap in the case of threshold algorithm (since each added pair will be visited twice). And that will also be a factor leading to an increased time with respect to larger values of K.