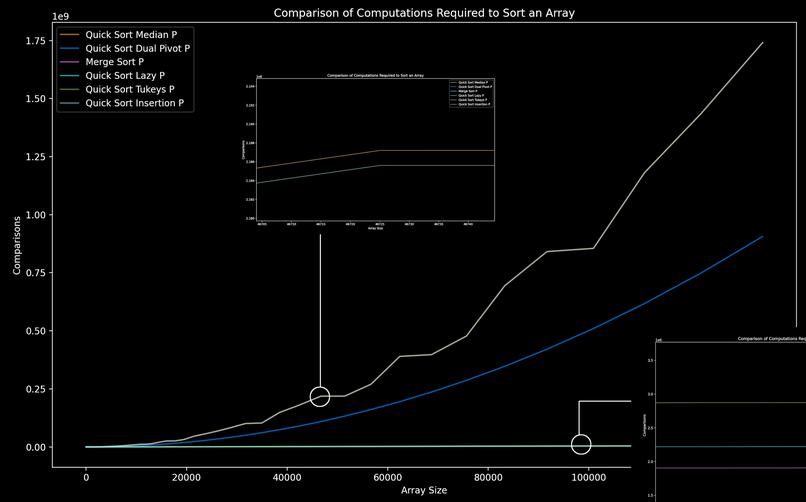
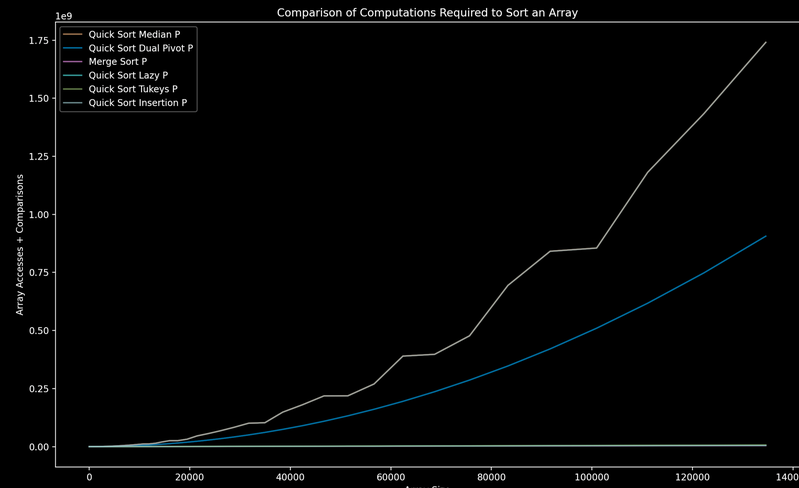
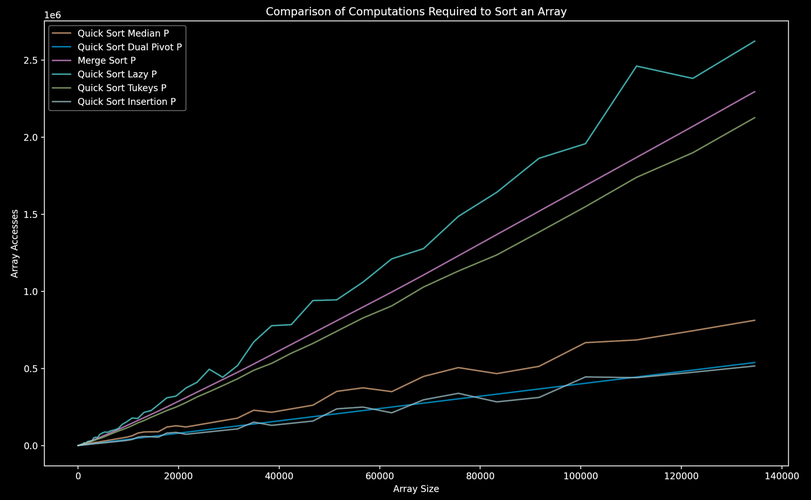
Recursive Sort Algorithm Performance on Arrays with Partially Sorted Data



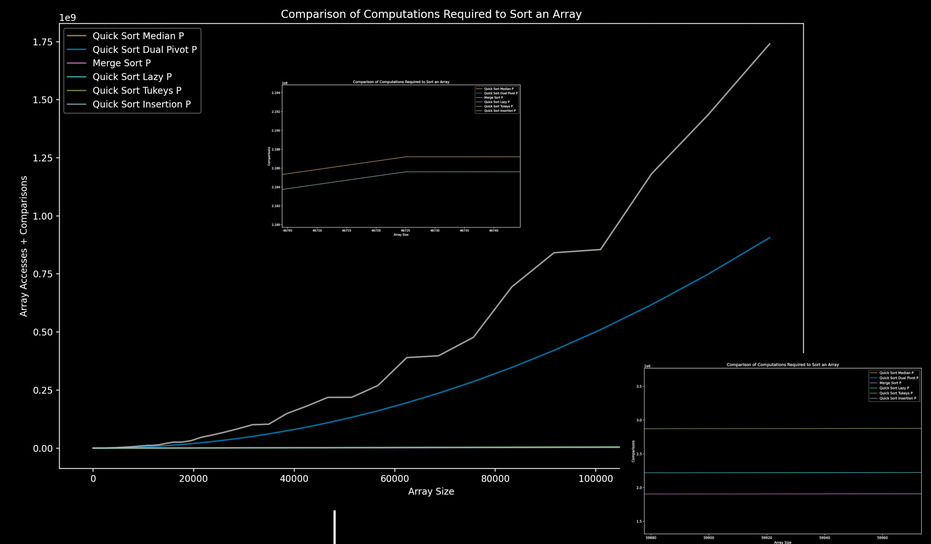
When sorting partially sorted data, the Median of 3 algorithm and Insertion degrade to O(n^2) performance with regard to comparisons performed. Dual pivot degrades to O(n^2) as well , but at a slower rate compared to the two aformentioned algos. Tukey's, Lazy (random pivot), and Merge all perform at O(NlogN).

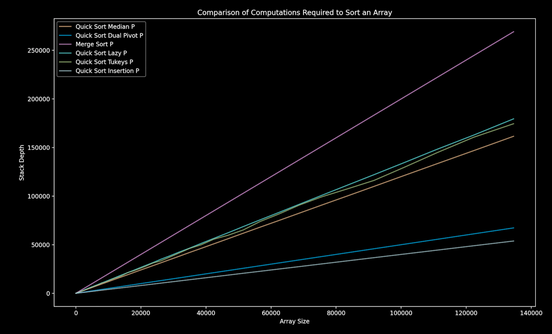


With regard to the combined data, the comparisons overpower all other data (1e9 for comparisons vs 1e6 for array accesses and less than 1 million for stack depths). As a result, the visual data is skewed entirely towards the comparison data.



With regard to array accesses, interestingly all algorithms perform at O(NlogN), despite some degrading to O(n^2) with comparisons. Median, Dual Pivot, and Insertion beging to outperform the others at a higher rate when array sizes grow larger.





As with the random data arrays, spacial complexity is is O(NlogN) for all algorithms. Quicksort Insertion outperforms all, which makes sense as this algorithm attempts to cut down on recursive calls when array sizes are very small (<~ 15 items), saving stack memory in the process. Merge sort performs the worst, as it will always recurse an array to its individual elements, creating array sizes of 1 for every item in an array.