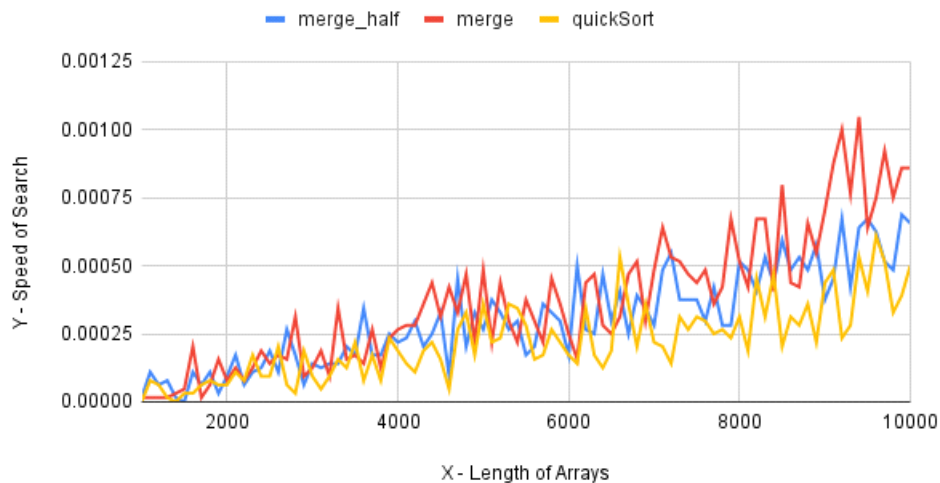


Sort Analysis

Thursday, March 9, 2023 3:09 PM

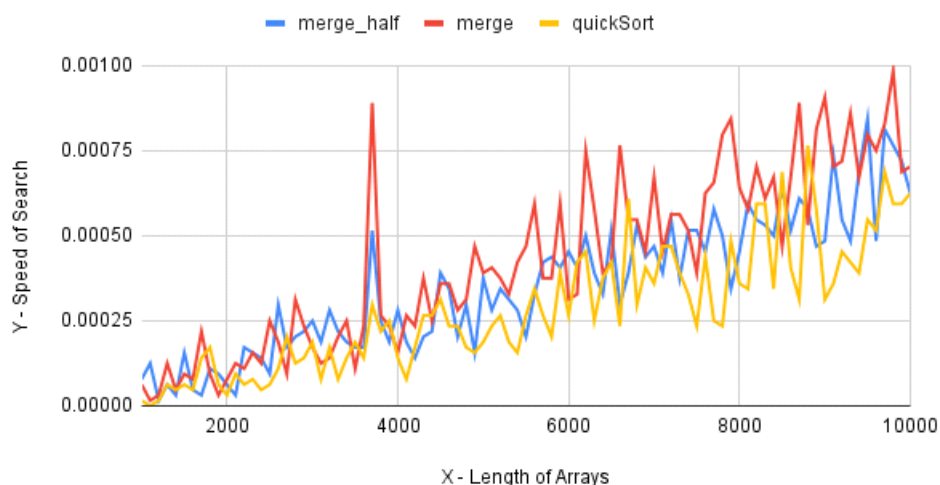
Improved Merges

Merge_half, Merge and QuickSort - Random



- This graph depicts the speed of different sorting algorithms for random inputs, in this case Merge, Merge_Half, and QuickSort. Comparing Merge and Merge_Half there should be little difference since our optimization focuses on saving space and not speed. Comparing both Merge sorts to QuickSort we should see a noticeable difference as QuickSort is a faster sorting algorithm.

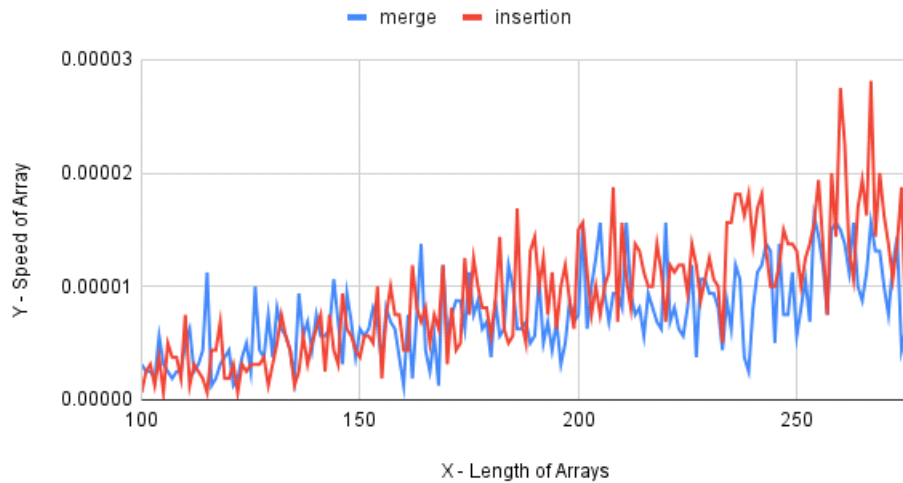
Merge_half vs Merge vs QuickSort - Ordered



- This graph depicts the speed of different sorting algorithms for ordered inputs, in this case Merge, Merge_Half, and QuickSort. Comparing Merge and Merge_Half there should be little difference since our optimization focuses on saving space and not speed. Comparing both Merge sorts to QuickSort we should see a noticeable difference as QuickSort is a faster sorting algorithm.

Threshold Section

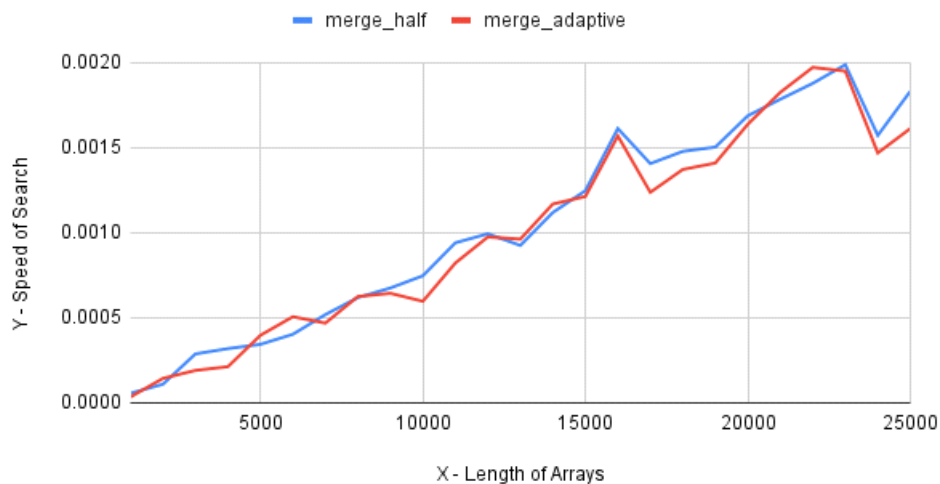
Merge vs Insertion



- This graph depicts the speeds for merge and insertion sort algorithms, the lengths of the arrays were much smaller to find the threshold where insertion began to slow down compared to insertion. As you can see insertion begins to take longer when the length of the array's reaches around 170 - 180. This let me assume that the threshold would be around that length.

Merge_Half vs Merge_Adaptive

Merge_half vs Merge_Adaptive

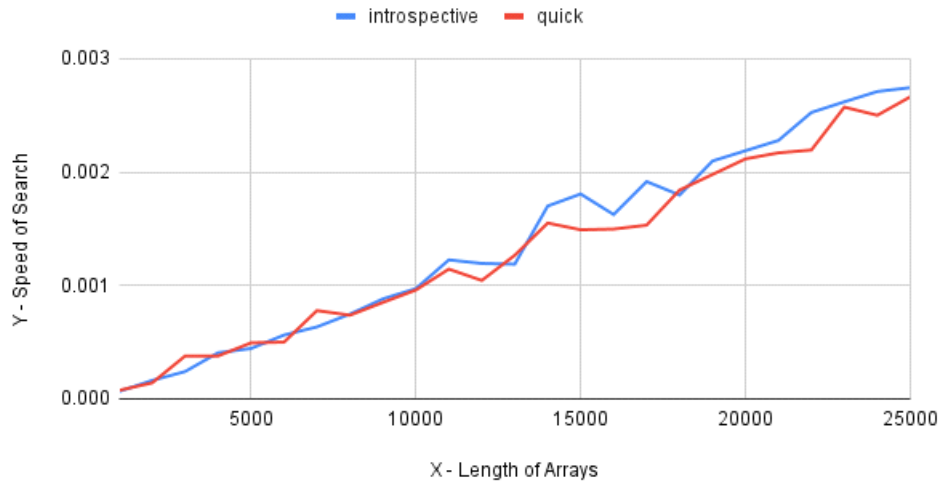


- This graph compares Merge_Half vs Merge_Adaptive sorting algorithms, merge_adaptive should be slightly faster than merge half since merge_adaptive switches algorithms from insertion to merge when a certain threshold is reached. As you can see most of time merge_adaptive is slightly faster

than merge half. Merge_Adaptive uses the same merge as merge_half so there should not be a big difference between the two algorithms.

IntrospectiveSort vs QuickSort - Random

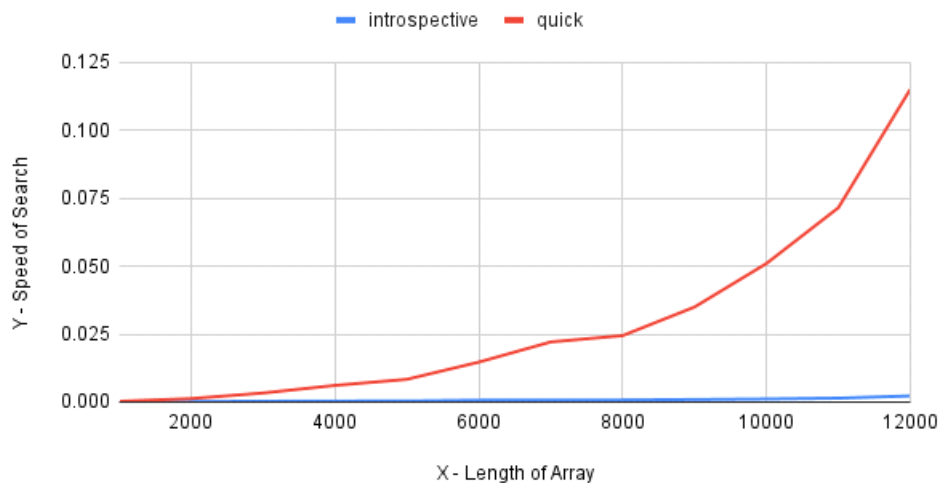
IntrospectiveSort vs QuickSort



- This graph compares Introspective sort vs Quick sort, quick sort should be faster than introspective sort when there is random inputs. Quick sort should be faster than introspective sort when there is random inputs because quick sort has a lower constant factor, as well fewer instruction per element. This graph shows that for quick sort as the speed of the search is faster for larger arrays with random inputs.

IntrospectiveSort vs QuickSort - Pathological

Introspective vs Quick



- This graph compares Introspective sort vs Quick sort, introspective sort should be faster than quick when there is pathological input. Introspective sort should be faster than introspective sort when there is pathological inputs because introspective sort is designed to switch algorithms when it detects

a recursion depth that exceeds a certain threshold. This prevents quicksort's worst case scenario. This graph shows this very clearly as introspective sort maintains a very linear slope while quick sort begins to slow down exponentially.