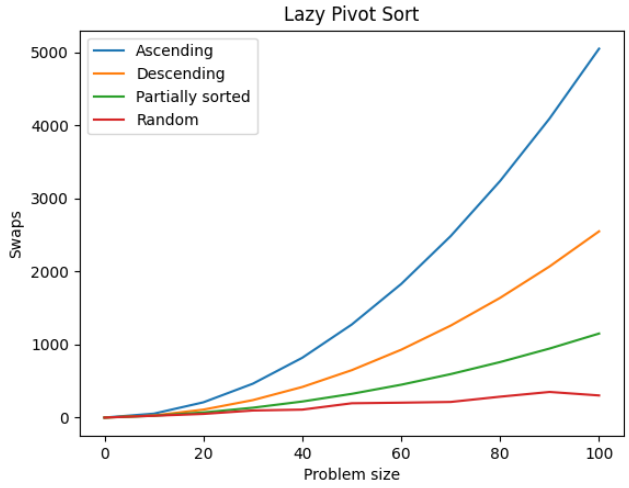
Experiment 1



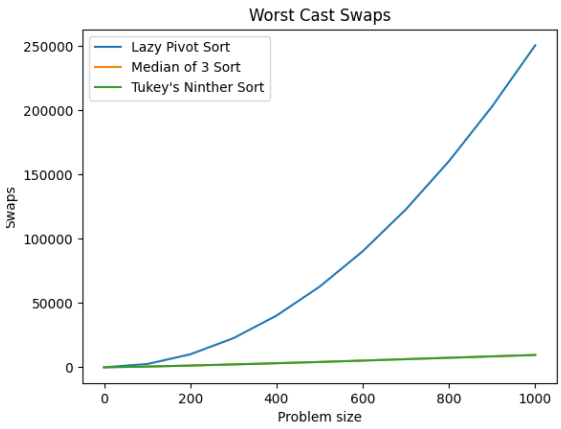
The Lomuto partition always starts with the last element as the pivot, which in an already sorted array is the largest element. The last element remains at the end and is swapped with itself.

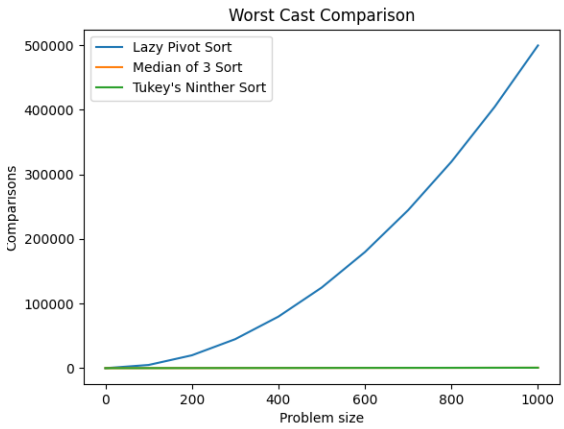
The next sublist to be partitioned is size n-1, continuing for subsequent sublist. Because the sublist is only one element smaller than the original sublist, the recursion depth becomes O(n). Each recursive call is O(n) (due to the partition loop), leading to O(n) * O(n) = $O(n^2)$ total swaps (since at least one swap is always performed at the end of the partition).

If the partitioning scheme were perfectly balanced, you'd expect O(n log n) comparisons and O(n) swaps for a fully sorted vector. However, because the pivot selection leads to unbalanced partitions, the number of swaps grows quadratically with the input size.

When the input is randomly sorted, the pivot is equally likely to be small, medium, or large, leading to roughly balanced partitions. Each recursive call splits the array into two roughly equal halves, resulting in O(log n) recursion depth.

Experiment 2

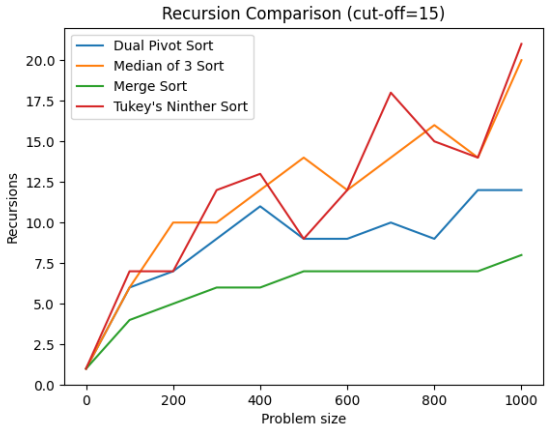




The median sorts use a better pivot selection process, resulting in better performance. The pivot is more likely to create balanced partitions, which should divide the list and each subsequent sublist into two equal size parts. If the happens the recursive calls with be made O(n log(n)) times.

The descending list will require every element to be moved the maximum number of places. The Lazy Pivot Sort will be O(n^2) because it has a naive pivot selection, which can be at one end of the list, resulting in sub optimal partitioning.

Experiment 3



Dual Pivot and Merge Sort have a smaller depth of recursion because they divide the list into more partitions than Tukey's Ninther Sort or Median of 3 sort. This means that for the same size n list, the base case will be reached faster as n is divided by a larger number, in this case 3 > 2.

The cut-off insertion reduces the number of total recursive calls for all sorts. The base case becomes the insertion limit, which if greater than one, will stop recursion calls early.