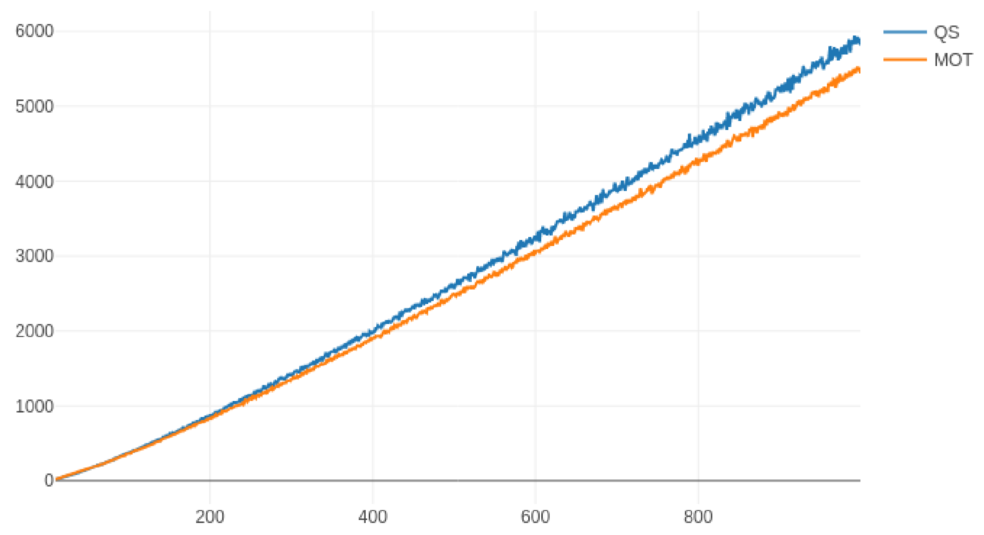
Jalil Farid
Programming Assignment 2

1. To produce an implementation of median of three partitioning, one can simply add a step to the quicksort procedure that chooses the pivot as the last element, where before this step, one substitutes the median of the first, middle, and last elements. See contained code for actual implementation.
2. Quicksort has a running time of $O(n * lg(n))$. Median of three partitioning is also $O(n * lg(n))$, whos proof follows. Each call of the algorithm first selects the median out of the first, middle, and last elements. Since the size of this median is fixed, take it to be constant. As the pseudocode shows, this step is added to each step of the quicksort algorithm, so each call to quicksort, which is known as $O(n * lg(n))$ by Master Theorem, has an additional step of constant time, adding an $O(n * lg(n))$ term to the complexity, so median of three's time complexity must be $O(n * lg(n)) + O(n * lg(n)) \approx O(n * lg(n))$.
3. On an already sorted input set, it is well known that quicksort degrades to $O(n^2)$ due to the recurrence obtained by choosing the last element as the source of the pivot, and calling quicksort again on a subproblem of size 0 and n-1, which generates as a recurrence relation $T(n) = O(n) + T(n-1)$. For an already sorted list, the median of three method diverts this worst case, choosing the median out of the first middle and last elements, giving a perfectly balanced subproblem. Using recurrence relations, and adding the constant term for finding the median of 3 elements, we have $T_{mot}(n) = O(1) + O(n) + 2T(n/2) = O(n) + 2T(n/2)$ which is $O(n * lg(n))$, a significant improvement in the worst case of quicksort when compared to it's running time of $O(n^2)$.
4. Code is included in the folder for implementations of both quicksort and median of three quicksort. In all code, each swap! operation adds to a stored count value, allowing an easy count of each exchange, which is computed for recursive calls as well. To show the improvement, for each k between i:j, 100 random permutation samples of integers from 1 to k is generated, and for both quicksort and median of three, the counts of the exchanges when applied to each sample is then averaged. Below is a plot showing the algorithm improvement.

Jalil Farid
Programming Assignment 2


QS
MOT