Assignment 5
Jovan Shandro

Problem 5.1

a) Runnig the three versions we can see that out of the three methods of partitioning, Hoare's is the slowest , then comes Lomuto's and then the fastest is the ' Median of three ' approach to partitioning. The third one performs better than the first two because it chooses the proper pivot depending on the array that is to be sorted. The first two always choose the pivot to be the first element that is why they perform worse compared to the median of three approach as the pivot is not chosen in any special way. As we can see by running Quicksort.java, the computation times in nanoseconds prove the above statements

Problem 5.2

b) We start by determining the best case.
Same as in the case of having one pivot which was discussed in class, the best case is reached when all 3 partitions have the same size. And since the other part of the function without the recurrences takes Θ(n), we get the recurrence
T(n) = 3T(n/3) + Θ(n). Using the master method for a = b = 3 and f(n) = Θ(n), $n^{\log_b a} = n^1 = n$ and as $f(n) = \Theta(n)$ by second case of the master method we get that T(n) = Θ(nlgn);
To consider the worst case, out of the three partitions 2 of them may have size one and the other n-2 thus we get the recurrence similar to the case with only one pivot :
T(n) = 2T(0) + T(n-2) + Θ(n) ; since T(0) is constant the formula becomes :
T(n) = T(n-2) + Θ(n) and calling it recursively we get T(n) = T(n-4) + 2 Θ(n)
= T(n-6) + 3 Θ(n) = ... = $\sum_{k=1}^{\lfloor n/2 \rfloor} \Theta(n) = \Theta(n^2)$. Thus similarly to the one pivot example, despite of course having differences in the constants of the asymptotic bounds, the time complexities are $\Theta(n^2)$ in worst case and Θ(nlgn) in best case scenario.

Problem 5.3

We first check the upper and then the lower bound:
$\log n! = \sum_{k=1}^{n} \log k \leq \sum_{k=1}^{n} \log n = n * logn$ thus lg(n!) = O(n*lgn).
Now for the lower bound we have :
$\log n! = \sum_{k=1}^{n} \log k \geq \sum_{k=n/2}^{n} \log k \geq \sum_{k=n/2}^{n} \log n/2 = n/2 * \log n/2$ thus lg(n!) = Ω (n*lgn)
So lg(n!) = Θ(n*lg(n))

References: https://en.wikipedia.org/wiki/Quicksort