Donovan Guelde
CSCI-3104
HW 3

1. $T(A) = 4T(n/4) + n \rightarrow O(n \log n)$

   $T(B) = 2T(n-1) + C \rightarrow O(n^2)$

   $T(C) = 9T(n/3) + n^2 \rightarrow O(n^2 \log n)$

   Choose algorithm A as the fastest

2. Via merge sort: dividing the array into terminal sub-arrays is M/2 operations

   combining the terminal sub-arrays into 1 array is M/2 operations

   Total comparisons made during recombining of arrays is n operations

   Total operations = $M/2 + M/2 + n = O(M+n)$

3. a. Time complexity = $O(k^2n)$

   b. A more efficient method would be to apply an altered merge sort to the arrays. Divide the arrays into sub arrays until they are of size 2. At this point, there is no need to compare and sort the 2-member arrays, as the parent arrays were sorted. From this point, recombine the arrays in the same manner as merge sort. The run time would be $O(k \log kn)$.

4.

Symbol	Start Date	End Date	Buy Date	Sell Date	Profit
COP	9/18/2014	9/18/2015	10/15/2014	11/21/2014	$10.68
NFX	9/18/2014	9/18/2015	1/14/2015	5/5/2015	$17.36
COH	9/18/2014	9/18/2015	11/5/2014	03/02/2015	$10.51

"On my honor, as a University of Colorado at Boulder student, I have neither given nor received unauthorized assistance."