1. Big O Problem List:

a) $T(n) = n^2 + 3n + 2 \implies O(n^2)$

b) $T(n) = (n^2 + n)(n^2 + \pi/2) \implies O(n^4)$

c) $T(n) = 1 + 2 + 3 + \ldots + n - 1 + n = n(n+1)/2 \implies O(n^2)$

d) $T(n) = 1^2 + 2^2 + 3^2 + \ldots + (n-1)^2 + n^2 = n(n+1)(2n+1)/6 \implies O(n^3)$

e) $T(n) = 10 \implies O(1)$

f) $T(n) = 10100 \implies O(1)$

g) $T(n) = n + log(n) \implies O(n)$

h) $T(n) = 12log(n) + n - 400 \implies O(n)$

i) $T(n) = (n+1) * log(n) - n \implies O(nlog(n))$

j) $T(n) = n^4 + 3n^2 + 2n \implies O(n^4)$

2. Find Big-O complexity for the following methods:

Method 1 : insertSort() $\implies O(n^2)$
Method 2 : allEvensUnder() $\implies O(limit)$, *or* $O(n)$

3. ArrayList Questions:

a) $O(1)$
b) $O(n)$
c) *There is an* $O(n)$ *time complexity because when removing an element, all of the others need to be shifted to fill the gap.*
d) $O(1)$
e) *The worst – case is* $O(1)$ *without reallocation. If reallocation is needed, the complexity would be* $O(log(n))$
f) *An ArrayList is appropriate when adding elements to the end of the list and accessing an element from a specific index,*

4. MasterArrayList Method Time Complexity:

unique() $\implies O(n^2)$
allMultiples() $\implies O(n)$
allStringsOfSize() $\implies O(n)$
isPermutation() $\implies O(nlog(n))$
tokenize() $\implies O(n)$
removeAll() $\implies O(n)$