Question 1

1 (a and b)

Algorithm thirdLargestByThreeLoops(arr)	#operations
firstIdx <- 0	1
secondIdx <- 0	1
thirdIdx <- 0	1
for int i <- 0 to arr.length do	2 + n
if arr[i] > arr[firstIdx] then	3n
firstIdx <- i	n
for int i <- 0 to arr.length do	2 + n
if i != firstIdx && arr[i] > arr[secondIdx] then	4n
secondIdx <- i	n
for int i <- 0 to arr.length do	2 + n
if i != firstIdx && i != secondIdx && arr[i] > arr[thirdIdx] then	5n
thirdIdx <- i	n
return arr[thirdIdx]	1
Total	**18n**

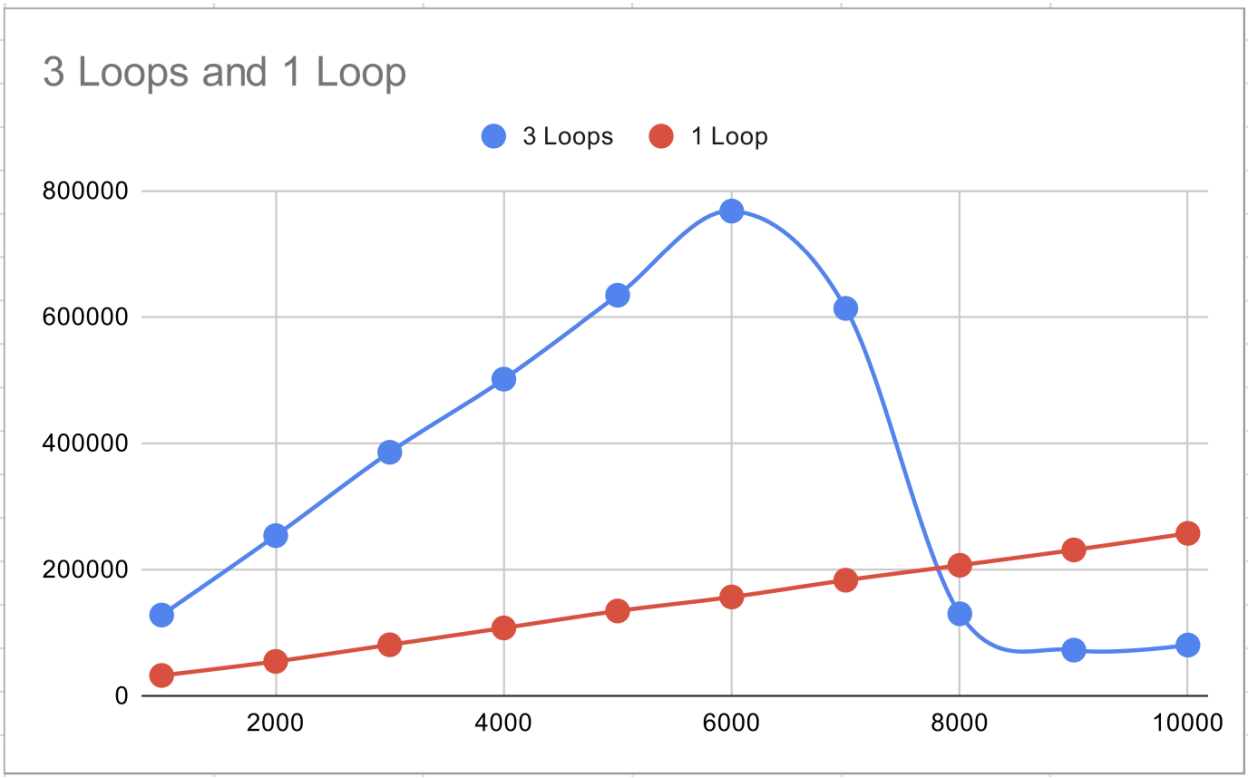
1 (c) Algorithm 1 by three loops: time complexity: O(n), space complexity O(n). Used extra 2 loops to determine the third max. Speed is slower than Algo2 because it runs two additional n times for loops.

2 (a and b)

findFirstThreeMax(A,n)	#operations
max<-minInt	1
preMax<-minInt	1
prePreMax<-minInt	1
for i<-0 to n-1 do	2 + n
if A[i]>prePreMax then	2n
if (A[i]>=max) then	2n
prePreMax = preMax	1n
preMax = max	1n
max = A[i]	2n
else if (A[i]>=preMax && A[i]<max) then	4n
prePreMax = preMax	1n
preMax = A[i]	2n
else	
prePreMax = A[i]	2n
Total	**18n**

2 (c) Algorithm 2 by 1 loop: time complexity: O(n) and space complexity is also O(n). Only used n times for loop to determine the third max. Speed is faster than an algorithm 1.

1 and 2 (d)



Question 2

10, 1	$\Theta(1)$
log(log(n))	$\Theta(\log(\log(n)))$
$n^{1/3}\log n$	$\Theta(n^{1/3}\log n)$
$n^{1/2}\log n$	$\Theta(n^{1/2}\log n)$
logn, ln n	$\Theta(\log n)$
$\log n^n$, nlogn	$\Theta(n\log n)$
$n^{1/k}(k>3)$	$\Theta(n^{1/k})$

$n^{1/3}$	$\Theta(n^{1/3})$
$n^{1/2}$	$\Theta(n^{1/2})$
n^2	$\Theta(n^2)$
n^3	$\Theta(n^3)$
$n^k (k>3)$	$\Theta(n^k)$
2^n	$\Theta(2^n)$
3^n	$\Theta(3^n)$
$n!$	$\Theta(n!)$
n^n	$\Theta(n^n)$