

## Questions 1 : Find Third Max

Algorithm 1 :

**findThirdMax(A)**

```

len ← length(A)
if len < 3 then
    return MIN_VALUE

first ← A[0]
firstIndex ← 0
secondIndex ← 0

for i ← 1 to len-1 do
    if A[i] > first then
        first ← A[i]
        firstIndex ← i

second ← MIN_VALUE

for i ← 0 to len -1 do
    if i != firstIndex AND A[i] > second AND A[i] <= first then
        second ← A[i]
        secondIndex ← i

third ← MIN_VALUE
for i ← 0 to len -1 do
    if i != firstIndex AND i != secondIndex AND A[i] > third AND A[i] <= second
then
        third ← A[i]

return third

```

**Numbers of operations =  $23n + 8$**

Algorithm 2 : **findThirdMax(A)**

	# Operations
max ← MIN_VALUE	1
preMax ← MIN_VALUE	1
prePreMax ← MIN_VALUE	1
for num in input	$n + 1 + n(\text{increments})$
if num > max then	n
prePreMax ← preMax	n
preMax ← max	n
max ← num	n
else if num > preMax then	n
prePreMax ← preMax	n
preMax ← num	n
else if num > prePreMax then	n
prePreMax ← num	n
return prePreMax	1
	<b>Total = <math>6n + 5</math></b>

$O(n)$

Algorithm 3 :  
getThirdMax(A)

```
len ← length(A)
if len = 0 then return MIN_VALUE
A.sort()
if len < 3 then return A[len - 1]
else return A[len-3]
```

$O(n \log n)$

Question 2.

Function	$\Theta$ -Notation
1, 10	$\Theta(1)$
$\log(\log n)$	$\Theta(\log \log n)$
$\log n$	$\Theta(\log n)$
$\ln n$	$\Theta(\log n)$
$n^{(1/k)}$ ( $k > 3$ )	$\Theta(n^{(1/k)})$
$n^{(1/3)}$	$\Theta(n^{(1/3)})$
$n^{(1/3)} \log n$	$\Theta(n^{(1/3)} \log n)$
$n^{(1/2)}$	$\Theta(n^{(1/2)})$
$n^{(1/2)} \log n$	$\Theta(n^{(1/2)} \log n)$
$n \log n$	$\Theta(n \log n)$
$n^2$	$\Theta(n^2)$
$n^3$	$\Theta(n^3)$
$n^k$ ( $k > 3$ )	$\Theta(n^k)$
$2^n$	$\Theta(2^n)$
$3^n$	$\Theta(3^n)$
$n!$	$\Theta(n!)$
$n^n$	$\Theta(n^n)$