**W1D2**

1. **Question 1**
2. Algorithm1

**Algorithm** *findThirdLargest(A, n)*

**Input** array **A** of **n** Integers

**Output** thirdLargestElement of **A**

firstMax 🡨 MIN\_VALUE

secondMax 🡨 MIN\_VALUE

thirdMax 🡨 MIN\_VALUE

firstMaxIndex 🡨 0

secondMaxIndex 🡨 0

**for** i 🡨 0 **to** n-1 **do**

currentElement 🡨 A[0]

**if** firstMax < currentElement **then**

firstMax 🡨 currentElement

firstMaxIndex 🡨 i

**for** i 🡨 0 **to** n-1 **do**

currentElement 🡨 A[0]

**if** i != firstMaxIndex **then**

**if** secondMax < currentElement **then**

secondMax 🡨 currentElement

secondMaxIndex 🡨 i

**for** i 🡨 0 **to** n-1 **do**

currentElement 🡨 A[0]

**if** i != secondMaxIndex **then**

**if** thirdMax < currentElement **then**

thirdMax 🡨 currentElement

**return** thirdMax

1. **Algorithm2**

thirdLargest(A,n)

**Input** array **A** of **n** Integers

**Output** thirdLargestElement of **A**

max 🡨 MIN\_INTEGER, preMax 🡨 MIN\_INTEGER, prePreMax 🡨 MIN\_INTEGER

for I 🡨 0 to n – 1 do

cur 🡨 A[0]

**if** cur > max **then**

prePreMax 🡨 preMax

preMax 🡨 max

max 🡨 cur

**else if** cur > preMax **then**

prePreMax 🡨 preMax

preMax = cur

**else if** cur > prePreMax **then**

prePreMax 🡨 cur

**return** prePreMax

1. Worst case

Algorithm1: O(3n) = O(n)

Algorithm2: O(n)

1. Chart
2. **Question 2**

Consider the following functions to determine the relationships that exist among the complexity classes they belong.

10, 1, n3, n1/3, log(log n), n2, n1/2, logn , log nn, nk (k > 3), , n1/k (k > 3), nlogn, ln n, 2n, 3n, nn, n1/2 logn, n1/3 logn, n!.

|  |  |
| --- | --- |
|  |  |
| 10, 1 | O(1) |
| log(logn) | O(log(logn)) |
| Logn, lnn | O(Logn) |
| n1/k | O(n1/k) |
| N1/3 | O(N1/3) |
| N1/3logn | O(N1/3logn) |
| N1/2 | O(N1/2) |
| n1/2logn | O(n1/2logn) |
| Nlogn, lognn | O(Nlogn) |
| N2 | O(N2) |
| N3 | O(N3) |
| Nk | O(Nk) |
| 2n | O(2n) |
| 3n | O(3n) |
| N! | O(N!) |
| Nn | O(Nn) |