**W1D2 Assignments**

**Question 1**

**Algorithm 1**

|  |  |
| --- | --- |
| **Algorithm FindThirdMaxUsingThreeLoop(A, n)** | **#Operations** |
|  |  |
| max1 ← A[0] | 2 |
|  |  |
| for i ← 1 to n - 1 do | 1 + n |
| if A[i] > max1 then | 2(n-1) |
| max1 ← A[i] | 2(n-1) |
|  |  |
| max2 ← -∞ | 1 |
|  |  |
| for i ← 0 to n - 1 do | 1+n |
| if A[i] > max2 and A[i] < max1 then | 2(n-1) |
| max2 ← A[i] | 2(n-1) |
|  |  |
| max3 ← -∞ | 1 |
|  |  |
| for i ← 0 to n - 1 do | 1+n |
| if A[i] > max3 and A[i] < max2 then | 2(n-1) |
| max3 ← A[i] | 2(n-1) |
|  |  |
| return max3 | 1 |
|  | **Total 14n -4** |

**Algorithm 2**

|  |  |
| --- | --- |
| **Algorithm FindThirdMaxUsingOneLoop(A, n)** | **#Operations** |
|  |  |
| max ← -∞ | 1 |
| preMax ← -∞ | 1 |
| prePreMax ← -∞ | 1 |
| for i ← 0 to n - 1 do | 1+n |
| if A[i] > max then | 2(n-1) |
|  |  |
| prePreMax ← preMax | n-1 |
| preMax ← max | n-1 |
| max ← A[i] | 2(n-1) |
| else if A[i] > preMax and A[i] < max then | 4(n-1) |
|  |  |
| prePreMax ← preMax | n-1 |
| preMax ← A[i] | 2(n-1) |
| else if A[i] > prePreMax and A[i] < preMax then | 4(n-1) |
|  |  |
| prePreMax ← A[i] | 2(n-1) |
|  |  |
| if prePreMax > 0 then | 1 |
| return prePreMax | 1 |
| else |  |
| return max | 1 |
|  | **Total 18n-10** |

**Algorithm 3**

|  |  |
| --- | --- |
| **Algorithm FindThirdMaxUsingOrderedDictionary(A, n)** | **#Operations** |
| Create an empty ordered dictionary | 1 |
| for each element in A do | n |
| if element is already in the dictionary, then | log n |
| Increment the count of the element in the dictionary by 1 | log n |
| else |  |
| Add the element to the dictionary with a count of 1 | log n |
|  |  |
|  |  |
| keys ← list of keys in descending order from the dictionary | n |
|  |  |
|  |  |
| if length(keys) ≥ 3 then | 2 |
| Return the third element in the list of descending keys (keys[2]) | 2 |
| else |  |
| Return the first element in the list of descending keys (keys[0]) | 2 |
|  |  |
| End | Total 2n log n +7 |

|  |  |  |
| --- | --- | --- |
| **Algorithms** | **Steps** | **Time Complexity** |
| Algorithm 1 | 14n -4 | O(n) |
| Algorithm 2 | 18n-10 | O(n) |
| Algorithm 3 | 2n log n + 7 | O(n log n) |

**Question 2**

|  |  |
| --- | --- |
| 10 ,1 | Θ (1) |
| log(log n) | Θ log(log n) |
| log n , ln n | Θ (log n) |
| n ^ 1/k (k > 3) | Θ (n ^ 1/k) |
| n ^ 1/3 | Θ(n ^ 1/3) |
| n ^ 1/3 logn | Θ(n ^ 1/3 logn) |
| n ^ 1/2 | Θ(n ^ ½) |
| n ^ 1/2 logn | Θ(n ^ 1/2 logn) |
| n | Θ (n) |
| nlogn , log nn | Θ(nlogn) |
| n^2 | Θ (n^2) |
| n^3 | Θ (n^3) |
| 2^n | Θ(2^n) |
| 3^n | Θ(3^n) |
| n! | Θ(n!) |
| n^n | Θ( n^n) |