Question 1

Algorithm 1: Pseudo Code and Cost

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Method1(A) | | | | | | | | |  |  |  |  |  |
| Input array A of integer | | | | | | |  |  |  |  |  |  |  |
| Output thirdLargestNumber of array | | | | | | | | | |  |  |  |  |
| Max <- A[0] | | | |  |  |  |  |  |  |  |  |  | 2 |
| count <- 1 | | |  |  |  |  |  |  |  |  |  |  | 1 |
| For i<- 1 to n-1 do | | | | |  |  |  |  |  |  |  |  | 1+n |
|  | If A[i] > Max then | | | | |  |  |  |  |  |  |  | 2(n-1) |
|  |  | Max <- A[i] | | | |  |  |  |  |  |  |  | 2(n-1) |
|  |  | count <- 1 | | |  |  |  |  |  |  |  |  | 1(n-1) |
|  | If A[i] == Max then | | | | |  |  |  |  |  |  |  | 2(n-1) |
|  |  | Count <- Count + 1 | | | | |  |  |  |  |  |  | 2(n-1) |
| SecondMax <- A[0] | | | | | |  |  |  |  |  |  |  | 2 |
| If Count >=2 then | | | | |  |  |  |  |  |  |  |  | 1 |
|  | SecondMax <- Max | | | | | |  |  |  |  |  |  | 1 |
|  | Count <- Count – 1 | | | | |  |  |  |  |  |  |  | 2 |
| Else then | | |  |  |  |  |  |  |  |  |  |  |  |
| For i<- 1 to n-1 do | | | | |  |  |  |  |  |  |  |  | 1+n |
|  |  | If A[i] > SecondMax AND A[i]<Max then | | | | | | | | | | | 4(n-1) |
|  |  |  | SecondMax <- A[i] | | | | |  |  |  |  |  | 2(n-1) |
|  |  |  | Count <- 1 | | |  |  |  |  |  |  |  | 1(n-1) |
|  |  | If A[i] == SecondMax then | | | | | | |  |  |  |  | 2(n-1) |
|  |  |  | Count <- Count + 1 | | | | |  |  |  |  |  | 2(n-1) |
| ThirdMax <- A[0] | | | | |  |  |  |  |  |  |  |  | 2 |
| If Count>=2 then | | | | |  |  |  |  |  |  |  |  | 1 |
|  | ThirdMax <- SecondMax | | | | | | |  |  |  |  |  | 1 |
| Count <- Count – 1 | | | | |  |  |  |  |  |  |  |  | 2 |
| Else then | | |  |  |  |  |  |  |  |  |  |  |  |
| For i<- 1 to n-1 do | | | | |  |  |  |  |  |  |  |  | 1+n |
|  |  | If A[i] > ThirdMax AND A[i]< SecondMax then | | | | | | | | | | | 4(n-1) |
|  |  |  | ThirdMax <- A[i] | | | | |  |  |  |  |  | 2(n-1) |
|  |  |  | Count <- 1 | | |  |  |  |  |  |  |  | 1(n-1) |
|  |  | If A[i] == ThirdMax then | | | | | | |  |  |  |  | 2(n-1) |
|  |  |  | Count <- Count + 1 | | | | |  |  |  |  |  | 2(n-1) |
| Return thirdMax | | | | |  |  |  |  |  |  |  |  | 1 |

**Total: 34n-12**

**Worst case time complexity: O(n) = n**

Algorithm 2 :

Algorithm Method2(A)

Input array A of integer

Output thirdLargestNumber of array

max<-0

preMax<-0

prePreMax<-0

count<-0

for i<-0 to n-1 do

if A[i]>max then

max<- A[i]

else if A[i]>preMax AND max> arr[i] then

preMax<-A[i]

else if count==0 AND A[i]==max then

preMax<-A[i]

Increment count

else if preMax > A[i] AND A[i] > prePreMax then

prePreMax <- arr[i]

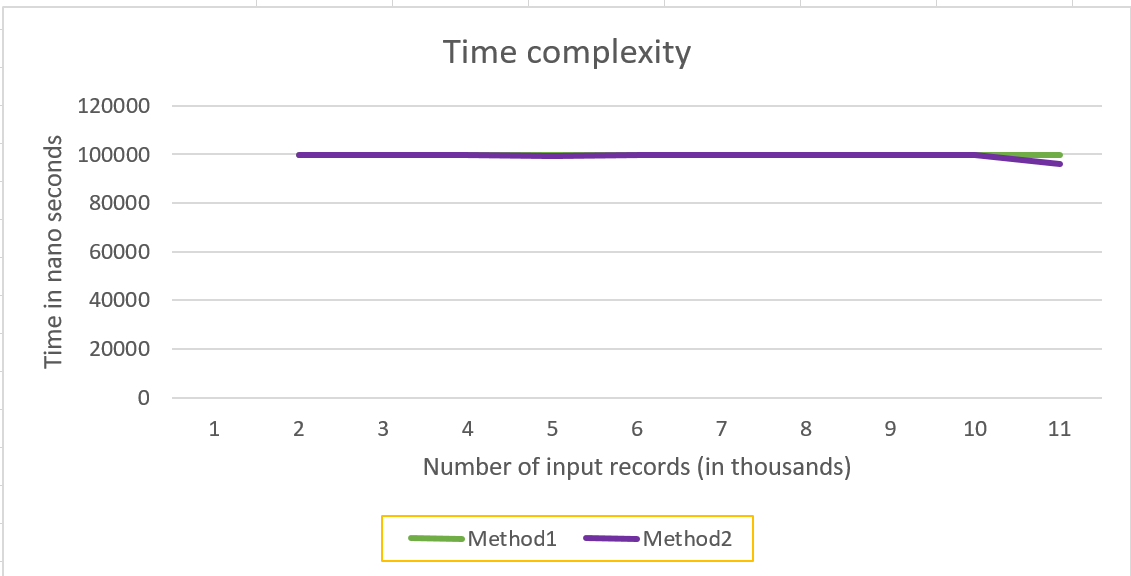
else if A[i]==preMax then

prePreMax <- arr[i]

return prePreMax

**Worst case time complexity O(n) = n**

**Graph**



**Question 2**

|  |  |
| --- | --- |
| 10,1 | ϴ(1) |
| log (log n) | ϴ(log(logn)) |
| log n, ln n | ϴ(logn) |
| n^1/k (k>3) | ϴ(n^1/k) |
| n^1/3 | ϴ(n^1/3) |
| n^1/3log(n) | ϴ(n^1/3log(n)) |
| n^1/2 | ϴ(n^1/2) |
| n^1/2log(n) | ϴ(n^1/2log(n)) |
| log(n^n), nlog(n) | ϴ(nlog(n)) |
| n^2 | ϴ(n^2) |
| n^3 | ϴ(n^3) |
| n^k (k>3) | ϴ(n^k) |
| 2 ^ n | O(2^n) |
| 3 ^n | O(3^n) |
| n! | O(n!) |
| n^n | O(n^n) |