**Problem Set 2**

**Answer 1:**

A=[initially sorted array]

N = A.length

Count = 0

While (count <= 15):

If item added:

Count++

If count == 15:

Count = 0

MergeSort(A , 0 , A.length ) //Code already stated in early //tasks and assignments

N = A.length

**Note:** We will be using Merge Sort algorithm to sort the arrays as it has a time complexity of O(nlogn) and it works better for large number of elements than the Insertion Sort or Selection Sort, as they have the time complexity of O(n2)

**Answer 2:**

For i=1 to length-1: n

count = 1 n-1

num = A[i] n-1

if num != ∞: n-1

for j=i+1 to length: (i=1 to i=n-1)∑n-i+1

if num == A[j]: (i=1 to i=n-1)∑n-i

count++ (i=1 to i=n-1)∑n-i

A[j] = ∞ (i=1 to i=n-1)∑n-i

Print(num+” “+count) n-1

Note: Its time complexity is O(n^2)

**Answer 3:**

Num = A[1] 1

count = 1 1

For i = 2 to length: n

If num == A[i]: n-1

Count++ n-1

Else: n-1

If count != 1: n-1

Print ( num + ” ” + count ) n-1

Num = A[i] n-1

Count = 1 n-1

Note: Its time complexity is O(n)

**Answer 4:**

MatrixMul(A ,B):

n = A.rows or A.coloumns 1

C = newMatrix of size n\*n 1

For i=1 to n: n+1

For j = 1 to n: n(n+1)

Cij = 0 n(n)

For k = 1 to n: n(n)(n+1)

Cij = Cij + Aik.Bkj n(n)(n)

Note: The time complexity is O(n^3)