Problem 5

a) Merging 1st and 2nd array of size n will take time of 2n steps, then merging the resultant array with a 3rd array will take 3n steps, and similarly, merging the resultant array with kth array will take kn steps. So the total time to merge all arrays = O(2n + 3n + 4n + …. + kn). Since, it is a series so total time $O(nk^2)$

b) A better algorithm:

```
def MergeArrays(A1, A2, A3…..Ak):
    k ← number of arrays
    H ← a min heap data structure
    sorted ← array to contain sorted element
    counter ← array of counters for each sorted array
    For i in range(k):
        Insert first elements, along with their array name, from each array into heap H
    While len(sorted) != nk:
        Remove an element from min heap
        Increase the counter of the array from which the element the belongs
        Add the element into sorted array
        Put the next element from that same array into the heap
```

While loop will run for O(nk) times whereas, heap will take O(logk) time to rearrange so in total this algorithm would take O(nklog(k)) time