Fibonacci Sequence:-

Code is uploaded to github - <https://github.com/IndrajaVallamkonda/CSE-5311/blob/main/Handson4/Fibonacci.java>  
  
for value 5 have got the following sequence  
fib(5) -> fib(4) -> fib(3) -> fib(2) -> fib(1) -> fib(0)  
-> fib(1)

-> fib(2)-> fib(1) -> fib(0)  
-> fib(3) -> fib(2) -> fib(1) -> fib(0)  
-> fib(1)

Problem 1:  
1. Code is uploaded to github link : <https://github.com/IndrajaVallamkonda/CSE-5311/blob/main/Handson4/SortedArraysMerge.java>  
  
2. Time complexity   
To take elements and store it in one array, we can assume a time complexity of O(N \* K)  
then have sorted the array based on merge sort. So overall time complexity is O(NK) + O(NK log (NK)) = O(NK log(NK))  
  
3. Improved approach  
Since each array is sorted could have used a min-heap that would return smallest element in constant time. Since the array is sorted we can take one element from each array and compare the first element of the array. After finding the smallest element we must take the next element from the same array and then compare again.  
Analysing this approach worst case time complexity is O(NK (log (K)).

Problem2:  
1. Code is uploaded to github link: <https://github.com/IndrajaVallamkonda/CSE-5311/blob/main/Handson4/RemoveDuplicateArrayElements.java>  
  
2. Time Complexity   
Array is traversal is done only once so time complexity would be O(n)  
  
3. Improved approach  
Could you think of any other approach as this is both space and time efficient, as only one array is used, have modified the input array to form the output. So I assume the approach is efficient.