Question 3

Zeal Liang

Z5325156

First, sort the array A incrementally using a merge sort algorithm.

We start by setting m to the largest integer in the array A. We need to traverse the array A with a different m. Starting with the first integer in array A, add one to the answer for each separation greater than or equal to m. Finally, check if you have chosen enough k, subtract $(m/2)/2$ from m if there are less than k, add $(m/2)/2$ to m if there are more than k, then re-traverse the array and update the answer。Until m is no longer divisible, return the answer.

The first step of sorting requires a time complexity of $O(n\ log\ n)$. The second step requires two nested loops, based on m and n respectively. The time complexity of the outer layer's m-based loop is $O(log\ m)$, and then the time complexity of each inner layer's n-based loop is $O(n\ )$, so the total time complexity is $O(n\ log\ m)$. Then because m is larger than n, the time complexity of these two steps together is $O(n\ log\ m)$.