1. Used mergeSort here. The merge function compares two sorted array and moves the pointer accordingly. Then returning a fully merged sorted array.

2. Uses the mergeSort function recursively to compare two values when array length is 0/1 to find out the maximum value.

3. Uses mergeSort plus added new variable to count inversion whenever the condition is not. Better complexity $O(n \log n)$ compared to original $O(n^2)$.

4. Compares [i]th value of array to squared array from [i+1] to last where the squared array is sorted in descending order to find the max $A[i] + A[j]^2$. Complexity $O(n \log n)$.

5. Quick sorting algorithm implemented here. Choosing a random pivot, less → array and a greater array. Then recursively calling less and greater array to solve the problem.

6. This modified Quick sort finds the k-th smallest element without full sorting. ~~Finding~~ during partitioning. If it finds k-1 smaller elements, the pivot is the answer. Otherwise, it searches appropriate subarray during the process.