First, we use merge sort to sort array A from minimum to maximum. We initialize a variable e = 0 to record final number of pairs. For i from 1 to n, we calculate x = S – A[i]. Then we use binary search to find x in sorted A and record index of x as m. If x is not in sorted A, then find the maximum element that is smaller than x and record index of x as m. the number of eligible A[j] for A[i] is n – m – 2. To record them, e = e + n – m – 2. After going through every i, e is our final answer. Merge sort cost run time O(nlogn). Calculating x cost O(1). Doing n binary search cost O(nlogn). So the time complexity is O(nlogn).