Lab 3

Task 1

To find $H_i > H_j$ for $i<j$ in efficient way we can use merge sort. As we know, merge sort's time complexity is $O(n \log n)$. So, if we use merge sort and while merging if we have to assign the second array's value in the temporary array then, we can say that $H_i > H_j$ for $i<j$, as second array's index is bigger than first array.

Task2

To find $A[i]+A[j]^2$ maximum value in a array with this formula in O(nlogn), we can use divide & conquer algorithm. We can divide the array into two until its length is 2. Then we would find two values from left array and right array. then, we would find max from left side and absolute max from right side. then, we can compare and return main maximum value.

Task3

We know, We have to choose pivot and shift every value to left, smaller than the pivot, and to right, every greater value than the pivot. By doing this we can sort an array. In my code, I choose last index as pivot and put every smaller value in left side of the pivot by swaping with two pointer

Task 4

I find the kth smallest value from a array by modifying the quicksort a little. After choosing a pivot, if the position is equals to k, then we know that that position's element is kth smallest value. However, ~~rather than~~ if we don't find the position equals, then if pos is greater, then we will recur in left subarray and if pos is smaller, then we will recur in right subarray.