Lab Assignment 03

22201936

Task 1

This problem is a straiglut forward implementation of merge sort. Here divide and conquer techrique is used. First divide the array untill sirgle ~~ler~~ element recusively, then merge the left and right portion. Time complexity $O(n \log n)$ and extra memory is used.

Task 2

In this problem, also used divide and conquered techrique to achive the time complexity of $O(\log n)$. Divided the array untill sirgle element. Then returning to the root compared and returned the maximum value.

Task 3

While merging, when a element cross another element, that implies, that is valid pair of $i < j$ and $H_i > H_j$. To keep track used global variable COUNTER.

Task 4

The function divide() is returning a tuple, which 0th index hold the maximum value in $i < j$ pair $H_i + [H_i]$. and the max of (i, j) to move further calculations. Here compared with absolute value before returning.

Task 5

This is a direct implementation of quick sort. Though quick sort doesn't take extra memory, but in the worst case Time complexity go $O(n^2)$

Task 6

To find the kth smallest value, in the partion algorithm if the chosen pivot is the kth element returned it. Other wise go to the left if k is smaller than pivot and right if k is greater. By doing it recursively program finds the answer.