COMP3121 21T2 Assignment 1 Q3

Written by Zheng Luo (z5206267)

The given array A can be sorted by using merge sort algorithm with time of $O(nlogn)$. L_i and U_i can be found by using binary search with time $O(logn)$, however it is necessary to check the preceding value for L_i in the array, whether the preceding value is same as L_i, if so, repeat the checking process until the preceding value is smaller than L_i, and apply same operation of U_i for checking consequent values. Then count the number of integers in between L_i and U_i with time $O(1)$. Since there are n pairs of integers of (L_i, U_i) given, hence the time complexity for using binary search becomes:

$$O(n) \times O(logn) = O(nlog(n))$$

In conclusion this algorithm satisfied the requirements and runs in time $O(nlog(n))$.