Explanation

Lab-02

Task-1

1. $O(n^2)$ means we can use 2 loops to solve the problem. According to the instruction, I use 2 for loop. 1 is outer loop and another is inner loop. I initialize a variable flag assigning the value False. This flag variable controls my inner loop. If flag is ~~fla~~ false then I execute my inner loop and write necessary output in output file otherwise I break the loop. After executing inner loop, if I get False in flag, then I write ~~IM~~ IMPOSSIBLE in output file as instructed.

Task-1

2. For O(n), I use 1 for loop which executes for N-1 times and when it meets flag == True, it breaks the loop. After that, if flag remains false it will write IMPOSSIBLE in output file.

Task-2

1. For O(n log n) I use merge sort. Before using merge sort, I add given two lists from Alice and Bob. Then I do sort the merge list using merge sort and write the output.

2. For O(n) I use 1 for loop which will execute for (N+M) times. I use 2 pointer for 2 lists. In for loop I compare the same index elements of two lists and then append them to

a new list according to ascending order. After the loop I check the lists for any element remaining. Then write the sorted list in output file.

Task-03:

First I store all tasks in a single list in list of lists format. After that I sort them by using bubble sort based on their endtime. Then I calculate the maximum tasks. I intialize count variable with intial value 1. and tasks variable with the value sorted list[0] which means the first task. Then I count the tasks which has equal or larger start time than the end time of previously done task. After calculation I write the total count of tasks I can do and which tasks are them in the output file.

Task-04:

Firstly I sort the list of tasks as same as task-03. But for O(n log n) I sort the list of tasks by merge-sort. After sorting I took 2 lists and counter variable named count = 1. Among 2 lists, 1 list is empty and another has value of first task. Then I calculate the maximum tasks by comparing the tasks which has equal or larger start time than the end time of any work done previously or which have equal start time and greater end time than previous task. After counting I write the output of maximum total number of tasks can be completed in the output file.

———— X ————