

### Task 01:

1)  $O(N^2)$  → for every element I run a nested loop which checks all the ~~element~~ succeeding elements. It checks and prints their positions if the sum is equal to the target sum.

2)  $O(N)$  → I run a while loop using two pointers from both sides. It adds up the two indices and checked it against the target sum. And changes the pointers as necessary.

### Task 02:

①  $O(N \log N)$  → I have sorted the two lists using mergesort.

②  $O(N)$  → Used a while loop and a pointer for each list. Whichever is small, gets added first.

### Task 03:

I first write the greedy algorithm. It sorts the list of tasks based on the summation of start and end time. Then takes the first element as first task. Then evaluates the rest to find the suitable next task. The process continues till all possible tasks are selected.

### Task 04:

The concept is the greedy algorithm here as well. But I created two separate lists for ease of choosing the first task efficiently. Then the base greedy algorithm does all the work.