Task 1 ($O(N^2)$)

Here I took two pointers mainly and a variable → flag = "False". This flag will work as a indicator to check. then I have taken two loops. First loop is the first pointer that is fixed for one iteration. The second one iterates the other element next from the first. Then they sum up. If my sum is equal to the sum (n) then the flag is changed and the loop breaks.

Task 1 (II) ($O(N)$)

It is similar to the previous one. But I used the reverse algorithm here. I used subtraction here. I made a dic and key as element and index as value. Then I subtracted from "key2". If the subtracted part was also in the dic. then I can say I found the solution.

Task 2(1)

Here I used merge sort and while loop. I checked first time and found alice list with bob list. If the element from "alice_nlist" is shorter than "bobnlist" it will automatically appended in the merge_list and vice versa. Then I used merged list fore all the remaining elements and got the result

Task 2(11)

I took a empty list fo solve the problem. Then used strip function to clean the list fully. I find out alice, bobs and newlist's length. Then I checked my conditions and accoreding to the conditions put the elements to the mergelist's empty position Then doing the merging again got the remaining elements

Task 3

First, I made a ~~li~~ empty list. Then I used the builtin-lambda function to sort the intervals based on their second element. Then I took my current_end variable = -1. So, in the first case the stopping time is greater than it. Then I updated the variable and got the final result

Task 4

I took two variables to ~~sto~~ take the start and ~~stop~~ time Then I iterated over the list ~~to~~ and found my current start and stop time. Then I used my "greedy algorithm" as named "fun" in a loop by ~~reg~~ the workers.