

1. Here first we took the desired in a variable and by iterating and adding when ever we find our desired value we save the last index tations and print and break the code (stop). $O(n^2)$

~~205~~
for part b we used key, value and dictionary to see if there's any match by using only one loop. Here we used flags to stop the loop while we are iterating inside the dictionary and find the wsum. Used logic $wsum - i = (\text{required num})$.

② a) For this one I used merge algorithm of merge sort as we have 2 sorted lists already which then called in mergesort to divide and conquer sorting. Which made its complexity $O(n \log n)$

b) For this I used loops that are not nested and for the extra parts that got left out I checked length and added onwards.

c) Here, I created a algorithm to schedule the work times according to the sorted array (that was sorted regarding end time of each work). and whenever it fits the low time criteria and always take the first one as it is already sorted. And finally adding them to a final array created before. Finally print the final array and its length according to desired output.

④ For this one I took the first line's values as n and m . And created a array with smaller arrays containing start and end times.

this time whenever a work is starting at one other works end time and is not already taken and not already in final array we add it to work, move ahead and add it to done works. And finally print (done) numbers at end and close the output file.