1. (a) Used nested loop to find the sum. If sum was not found then returned "IMPOSSIBLE". Two loops were used so complexity $O(n^2)$

(b) Used one loop only to compare the values. Took two pointers named left and right pointer starting from 0 and n-1 respectively. If the sum of two pointers array was smaller than targeted sum we moved left pointer by +1 and if the sum of two pointers array was greater than the targeted sum we moved the right pointer array by -1. Thus achieving a complexity of $O(n)$

2. (a) At first merged the two arrays then sorted using merge sort algorithm. So, complexity of $O(n \log n)$ achieved.

(b) Compared two arrays, each with separate pointer. Then added the values to a new array named sorted list in ascending order. Used one while loop here. Thus, $O(n)$ complexity was achieved.

3. Used greedy algorithm where the end time becomes starting time for the next task.

4. Same as problem 3. Plus, we used nonpower to loop through the tasks.