Lab1 Sum of K Clarification

I provide some examples, Math part. You have to figure out how to implement the algorithm in O(n) complexity.

An example of two numbers

For example, you have the sorted list below, and need to find 2 elements that sum is, for instance, is equal *8*. Assume that P1 and P2 are pointers that initially point to the first and the last numbers

1	5	7	8	12	13
P1					P2

You add P1 and P2: 1+13 > 8. So, move pointer P2 to the left. Now

1	5	7	8	12	13
P1				P2	

Now add 1+12, still 1+12 > 8, so again, move pointer P2 to the left

1	5	7	8	12	13
P1			P2		

Now add 1+8, still 1+8 > 8, so again, move pointer P2 to the left

1	5	7	8	12	13
P1		P2			

1+7 == 8 Yes! Bingo! We found our solution!

An example where there is no solution

But what if we do not have 7 in our list? And the original list is

1	5	6	8	12	13
P1					P2

We continue the same operations

1	5	6	8	12	13
P1				P2	

Now add 1+12, still 1+12 >8, so again, move pointer P2 to the left

1	5	6	8	12	13
P1			P2		

Now add 1+8, still 1+8 >8, so again, move pointer P2 to the left

1	5	6	8	12	13
P1		P2			

1+6 < 8, so we move pointer P1 to the right

1	5	6	8	12	13
	P1	P2			

5+6>8, move pointer P2 to the left

1	5	6	8	12	13
	P1, P2				

Now pointers P1 and P2 point to the same element - only one chance to get the solution is 5+5, but 5+5 is 10 and is not equal to 8. So, we proved by construction that there are no solutions for the given list and given "target" number 8.

You got the idea, how to build the algorithm? The key is that the list is sorted, it is why we can move pointers left and right.

Actually, since we have deal with the C++ array, pointers, in our case, can be indexes of the array.

An example where there are multiple solutions

Just for further clarification, let's consider one more array and sum of two numbers that is 13

Find all pairs, that their sum is 13

1	3	5	7	8	12	14
P1						P2

1+14 > 8, move P2 to the left

1	3	5	7	8	12	14
P1					P2	

1+12 == 13, Bingo! Solution is found, but what if we have more solutions?

If we move just one pointer, we violate equality, so we need to move both pointers the same time

1	3	5	7	8	12	14
	P1			P2		

3+8 < 13, so we move P1 to the right

1	3	5	7	8	12	14
		P1		P2		

5+8 == 13! Bingo! We found one more solution! Are you starting to like it?

Maybe we have even more solutions? Move both pointers:

1	3	5	7	8	12	14
			P1,P2			

Now, as pointers point to the same number 7, the only option is that our solution is 7+7, but 7+7 is 14, and it is not equal to 13.

At first glance, it is a failure but it, actually, is our success as we proved that problem has only two solutions 1+12 and 5+8, no more. We proved it by construction.

To remind again, the list is sorted. It is why we could move pointers (or indexes of the array), left and right. And all these calculations are O(n).