Finding an integer pair with given sum is a very familiar problem in algorithm community. The general problem is: given an array of integer numbers and a given sum, how do we find out a integer pair with given sum?

The easiest solution is to consider every possible pair and compare it with the given number. Time complexity of this approach will be O(n^2) which is too big if the array size is very big.

Most efficient solution is to applying an efficient sorting algorithm to sort the numbers first. You can use QuickSort which has average case time complexity of O(nlogn). Once the list is sorted, we can use a linear search on the array to find out the pair with a given sum. Use the following pseudo code.

1. initialize two pointer: **left** & **right** pointing to the first and last element of the sorted array
2. if array[**left**] + array[**right**] == **SUM** then return **true**
3. else if array[**left**] + array[**right**] < **SUM** then **left**++
4. else **right**– –
5. if **left** < **right** then go to step 2
6. else return **“no pair found”**

Why this algorithm works? The intuition is simple. The main technique is to move the left and right pointer. If array[**left**] + array[**right**] < **SUM** that means we have to take a bigger number from the array to form the given sum. So we move the left pointer in right direction to form a bigger sum. As the array is sorted in increasing order, by moving the left pointer we’ll get a bigger number and hence bigger sum. On the other hand if array[**left**] + array[**right**] > **SUM**, then we’ll have to take a smaller number to form the given sum. We do that by moving the right pointer to the left where we’ll get a smaller number.