

← Back To Course Home

# Grokking the Coding Interview: Patterns for Coding Questions

7% completed



Q Search Course

- Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)

## Triplets with Smaller Sum (medium)

We'll cover the following ^

- Problem Statement
- Try it yourself
- Solution
  - Code
  - Time complexity
  - Space complexity
- Similar Problems
  - Time complexity
  - Space complexity

### Problem Statement #

Given an array `arr` of unsorted numbers and a target sum, **count all triplets** in it such that `arr[i] + arr[j] + arr[k] < target` where `i`, `j`, and `k` are three different indices. Write a function to return the count of such triplets.

### Example 1:

[← Back To Course Home](#)

# Grokking the Coding Interview: Patterns for Coding Questions

7% completed



- ☒ Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- ☐ Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- ☐ Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- ☐ Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- ☐ Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/...

Input: [-1, 0, 2, 3], target=3

Output: 2

Explanation: There are two triplets whose sum is less than the target: [-1, 0, 3], [-1, 0, 2]



## Example 2:

Input: [-1, 4, 2, 1, 3], target=5

Output: 4

Explanation: There are four triplets whose sum is less than the target: [-1, 1, 4], [-1, 1, 3], [-1, 1, 2], [-1, 2, 3]

## Try it yourself #

Try solving this question here:

Java

Python3

JS

C++

```
1 import java.util.*;
2
3 class TripletWithSmallerSum {
4
5     public static int searchTriplets(int[] arr, int target) {
6         int count = 0;
7         // TODO: Write your code here
8         Arrays.sort(arr);
9         for(int i=0;i<arr.length;i++) {
10             int start = i+1,end = arr.length-1;
11             while(start<end) {
12                 if(arr[i]+arr[start]+arr[end]<target){
13                     count += end-start;
14                     start++;
15                 }
16                 else{
17                     end--;
18                 }
19             }
20         }
21         return count;
22     }
23 }
```








← Back To Course Home

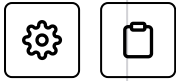
# Grokking the Coding Interview: Patterns for Coding Questions

7% completed 



-  Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
-  Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
-  Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
-  Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
-  Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/...

```
18     }
19     }
20     }
21     return count;
22 }
23 }
```



Test

Save \*

Reset



Show Results

Show Console



 2 of 2 Tests Passed

Result	Input	Expected Output	Actual Output	Reason
✓	searchTriplets([-1, 0, 2, 3], 3)	2	2	Succeeded
✓	searchTriplets([-1, 4, 2, 1, 3], 5)	4	4	Succeeded

3.574s

## Solution #

This problem follows the **Two Pointers** pattern and shares similarities with Triplet Sum to Zero  
(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/56795>)

[← Back To Course Home](#)

## Grokking the Coding Interview: Patterns for Coding Questions

7% completed



- ☒ Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- ☐ Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- ☐ Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- ☐ Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- ☐ Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/...

49973004288/). The only difference is that, in this problem, we need to find the triplets whose sum is less than the given target. To meet the condition  $i \neq j \neq k$  we need to make sure that each number is not used more than once.



Following a similar approach, first, we can sort the array and then iterate through it, taking one number at a time. Let's say during our iteration we are at number 'X', so we need to find 'Y' and 'Z' such that  $X + Y + Z < target$ . At this stage, our problem translates into finding a pair whose sum is less than " $target - X$ " (as from the above equation  $Y + Z == target - X$ ). We can use a similar approach as discussed in Triplet Sum to Zero

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/5679549973004288/>).

### Code #

Here is what our algorithm will look like:

Java

Python3

C++

JS

```
1 import java.util.*;
2
3 class TripletWithSmallerSum {
4
5     public static int searchTriplets(int[] arr, int target) {
6         Arrays.sort(arr);
7         int count = 0;
8         for (int i = 0; i < arr.length - 2; i++) {
9             count += searchPair(arr, target - arr[i], i);
10        }
11        return count;
12    }
13
14    private static int searchPair(int[] arr, int target) {
15        int count = 0;
16        int start = 0, end = arr.length - 1;
17        while (start < end) {
18            int sum = arr[start] + arr[end];
19            if (sum < target) {
20                count++;
21                start++;
22            } else {
23                end--;
24            }
25        }
26        return count;
27    }
28 }
```



[← Back To Course Home](#)

# Grokking the Coding Interview: Patterns for Coding Questions

7% completed



- Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/...

```
16 int left = first + 1, right = arr.length - 1;
17 while (left < right) {
18     if (arr[left] + arr[right] < targetSum) { //
19         // since arr[right] >= arr[left], therefore
20         // left and right to get a sum less than th
21         count += right - left;
22         left++;
23     } else {
24         right--; // we need a pair with a smaller s
25     }
26 }
27 return count;
28 }
29
30 public static void main(String[] args) {
31     System.out.println(TripletWithSmallerSum.search
```

Run

Save

Reset



## Time complexity #

Sorting the array will take  $O(N * \log N)$ . The `searchPair()` will take  $O(N)$ . So, overall `searchTriplets()` will take  $O(N * \log N + N^2)$ , which is asymptotically equivalent to  $O(N^2)$ .

## Space complexity #

The space complexity of the above algorithm will be  $O(N)$  which is required for sorting if we are not using an in-place sorting algorithm.

## Similar Problems #

**Problem:** Write a function to return the list of all such triplets instead of the count. How will the time complexity change in this case?

**Solution:** Following a similar approach we can create a list containing all the triplets. Here is the code - only the highlighted lines have changed:



[← Back To Course Home](#)

## Grokking the Coding Interview: Patterns for Coding Questions

7% completed



- Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/...

Java

Python3

C++

JS

```
1 import java.util.*;
2
3 class TripletWithSmallerSum {
4
5     public static List<List<Integer>> searchTriplets(
6         Arrays.sort(arr);
7         List<List<Integer>> triplets = new ArrayList<>();
8         for (int i = 0; i < arr.length - 2; i++) {
9             searchPair(arr, target - arr[i], i, triplets);
10        }
11        return triplets;
12    }
13
14    private static void searchPair(int[] arr, int targetSum, int first,
15        int left = first + 1, right = arr.length - 1;
16        while (left < right) {
17            if (arr[left] + arr[right] < targetSum) { //
18                // since arr[right] >= arr[left], therefore
19                // left and right to get a sum less than the targetSum
20                for (int i = right; i > left; i--)
21                    triplets.add(Arrays.asList(arr[first], arr[left], arr[i]));
22                left++;
23            } else {
24                right--; // we need a pair with a smaller sum
25            }
26        }
27    }
28
29    public static void main(String[] args) {
30        System.out.println(TripletWithSmallerSum.searchTriplets(arr, targetSum));
31        System.out.println(TripletWithSmallerSum.searchTriplets(arr, targetSum));
32    }
33 }
```

Run

Save

Reset



[← Back To Course Home](#)

# Grokking the Coding Interview: Patterns for Coding Questions

7% completed



- ☒ Solution Review: Problem Challenge 2  
(/courses/grokking-the-coding-interview/xl2g3vvrMq3)
- ☐ Problem Challenge 3  
(/courses/grokking-the-coding-interview/3wDJAYG2pAR)
- ☐ Solution Review: Problem Challenge 3  
(/courses/grokking-the-coding-interview/xoyL4q6ApNE)
- ☐ Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)
- ☐ Solution Review: Problem Challenge 4  
(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)

Another simpler approach could be to check every triplet of the array with three nested loops and create a list of triplets that meet the required condition.



## Time complexity #

Sorting the array will take  $O(N * \log N)$ . The `searchPair()`, in this case, will take  $O(N^2)$ ; the main `while` loop will run in  $O(N)$  but the nested `for` loop can also take  $O(N)$  - this will happen when the target sum is bigger than every triplet in the array.

So, overall `searchTriplets()` will take  $O(N * \log N + N^3)$ , which is asymptotically equivalent to  $O(N^3)$ .

## Space complexity #

Ignoring the space required for the output array, the space complexity of the above algorithm will be  $O(N)$  which is required for sorting.

[← Back](#)

[Next →](#)

Triplet Sum Close to Target (medium)

Subarrays with Product Less than a Ta...

☒ **Mark as Completed**

7% completed, meet the [criteria](#) and claim your course certificate!

[Claim Certificate](#)



Report  
an Issue



Ask a Question

([https://discuss.educative.io/tag/triplets-with-smaller-sum-medium\\_\\_pattern-two-pointers\\_\\_grokking-the-coding-interview-patterns-for-coding-questions](https://discuss.educative.io/tag/triplets-with-smaller-sum-medium__pattern-two-pointers__grokking-the-coding-interview-patterns-for-coding-questions))

[← Back To Course Home](#)

# Grokking the Coding Interview: Patterns for Coding Questions

7% completed



 *Search Course*

Solution Review: Problem Challenge 2

(/courses/grokking-the-coding-interview/xl2g3vvrMq3)

Problem Challenge 3

(/courses/grokking-the-coding-interview/3wDJAYG2pAR)

Solution Review: Problem Challenge 3

(/courses/grokking-the-coding-interview/xoyL4q6ApNE)

Problem Challenge 4

(/courses/grokking-the-coding-interview/Y5YDWzqPn7O)

Solution Review: Problem Challenge 4

(/courses/grokking-the-coding-interview/...