







Grokking the Coding Interview: Patterns for Coding Questions

7% completed

C

Search Course

IIILEI VIEW/INUOSQIIFLNINV)

Problem Challenge 2
(/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem
Challenge 2
(/courses/grokking-the-coding-

interview/xl2q3vxrMq3)

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

(/courses/grokking-the-coding-

(/courses/grokking-the-coding-

Solution Review: Problem

interview/xoyL4q6ApNE)

Problem Challenge 4

Problem Challenge 3

Challenge 3

Triplet Sum Close to Target (medium)

We'll cover the following

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

Problem Statement

Given an array of unsorted numbers and a target number, find a **triplet in the array** whose sum is as close to the target number as possible, return the sum of the triplet. If there are more than one such triplet, return the sum of the triplet with the smallest sum.

Example 1:

7% completed

Q Search Course

ITILET VIEW/INUOSQTIPLKINV)

Problem Challenge 2 (/courses/grokking-the-codinginterview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2 (/courses/grokking-the-coding-

interview/xl2q3vxrMq3)

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-

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

Output: 1

Explanation: The triplet [-2, 1, 2] has the closest sum to the target.

\$\$



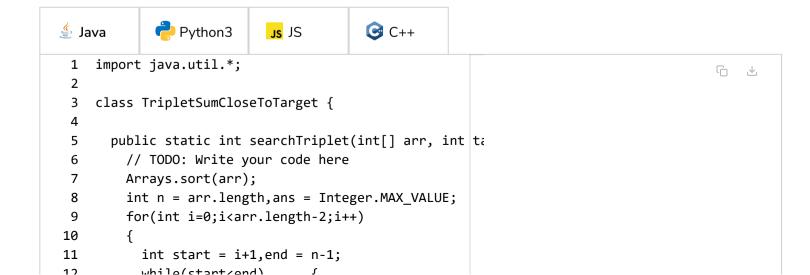
```
Input: [-3, -1, 1, 2], target=1
Output: 0
Explanation: The triplet [-3, 1, 2] has the closest sum to the target.
```

Example 3:

```
Input: [1, 0, 1, 1], target=100
Output: 3
Explanation: The triplet [1, 1, 1] has the closest sum to the target.
```

Try it yourself

Try solving this question here:



← Back To Course Home

Grokking the Coding Interview: Patterns for Coding Questions

7% completed

C

Q Search Course

ILITEL AIGN/IA002 ALLE VIANA

Problem Challenge 2 (/courses/grokking-the-codinginterview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2 (/courses/grokking-the-coding-

interview/xl2q3vxrMq3)

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

Solution Review: Problem Challenge 3 (/courses/grokking-the-codinginterview/xoyL4q6ApNE)

Problem Challenge 4 (/courses/grokking-the-coding-

```
int diff = targetSum-arr[i]-arr[start]-arr[
13
            if(diff==0){
14
              return targetSum-diff;
15
16
            ans = Math.min(ans, Math.abs(diff));
17
            if(diff>0){
18
              start++;
19
20
21
            else{
22
              end--;
23
24
25
26
        return targetSum-ans;
27
28
   }
```

Test Save * Reset []

X

Show Results Show Console

📋 3 of 3 Tests Passed

Result	Input	Expected Output	Actual Output	Reason
~	searchTriplet([-2, 0, 1, 2], 2)	1	1	Succeeded
~	searchTriplet([-3, -1, 1, 2], 1)	0	0	Succeeded
~	searchTriplet([1, 0, 1, 1], 100)	3	3	Succeeded

3.883s

← Back To Course Home

Grokking the Coding Interview: Patterns for Coding Questions

7% completed

Q Search Course

IIITEI VIEW/NOOSQIIPLKINV)

Problem Challenge 2 (/courses/grokking-the-codinginterview/YQ8N2OZq0VM)

Solution Review: Problem

Challenge 2

(/courses/grokking-the-coding-

interview/xl2g3vxrMq3)

Problem Challenge 3

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

Solution Review: Problem

Challenge 3

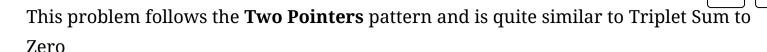
(/courses/grokking-the-coding-

interview/xoyL4q6ApNE)

Problem Challenge 4

(/courses/grokking-the-coding-

Solution



(https://www.educative.io/collection/page/5668639101419520/5671464854355968/56795 49973004288/).

We can follow a similar approach to iterate through the array, taking one number at a time. At every step, we will save the difference between the triplet and the target number, so that in the end, we can return the triplet with the closest sum.

Code

Here is what our algorithm will look like:

```
👙 Java
            Pvthon3
                            G C++
                                         JS JS
 1 import java.util.*;
                                                                                            Ψ,
 2
    class TripletSumCloseToTarget {
 4
       public static int searchTriplet(int[] arr, int ta
 5
         if (arr == null || arr.length < 3)</pre>
 6
           throw new IllegalArgumentException();
 7
 8
         Arrays.sort(arr);
 9
         int smallestDifference = Integer.MAX VALUE;
10
11
         for (int i = 0; i < arr.length - 2; i++) {
           int left = i + 1, right = arr.length - 1;
12
13
           while (left < right) {</pre>
             // comparing the sum of three numbers to the
14
             // so, we will try to find a target differe
15
             int targetDiff = targetSum - arr[i] - arr[]
16
             if (targetDiff == 0) // we've found a trig
17
               return targetSum - targetDiff; // return
18
19
```

Back To Course Home

Grokking the Coding Interview: Patterns for Coding Questions

7% completed

Q Search Course

IIIITEI NIEWINOOƏÇIIL FVINN

Problem Challenge 2 (/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem
Challenge 2
(/courses/grokking-the-coding-interview/xl2q3vxrMq3)

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

Solution Review: Problem Challenge 3 (/courses/grokking-the-codinginterview/xoyL4q6ApNE)

Problem Challenge 4 (/courses/grokking-the-coding-

```
20
            // the second part of the above 'if' is to
            if (Math.abs(targetDiff) < Math.abs(smalles</pre>
21
                || (Math.abs(targetDiff) == Math.abs(sn
22
23
              smallestDifference = targetDiff; // save
24
            if (targetDiff > 0)
25
              left++; // we need a triplet with a bigge
26
27
            else
              right--; // we need a triplet with a smal
28
29
30
        return targetSum - smallestDifference;
31
                                                                                    Reset
 Run
                                                                         Save
```

Time complexity

Sorting the array will take O(N*logN). Overall searchTriplet() will take $O(N*logN+N^2)$, which is asymptotically equivalent to $O(N^2)$.

Space complexity #

The space complexity of the above algorithm will be $\mathcal{O}(N)$ which is required for sorting.



7% completed, meet the <u>criteria</u> and claim your course certificate!

Claim Certificate

← Back To Course Home

Grokking the Coding Interview: Patterns for Coding Questions

7% completed

C

Q Search Course

ITILET VIEW/INUOSQTIPLKINV)

Problem Challenge 2 (/courses/grokking-the-coding-interview/YQ8N2OZq0VM)

Solution Review: Problem Challenge 2 (/courses/grokking-the-codinginterview/xl2g3vxrMq3)

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

Solution Review: Problem Challenge 3 (/courses/grokking-the-codinginterview/xoyL4q6ApNE)

Problem Challenge 4 (/courses/grokking-the-coding-

? Ask a Question



(https://discuss.educative.io/tag/triplet-sum-close-to-target-medium__pattern-two-pointers__grokking-the-coding-interview-patterns-for-coding-questions)



