

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n^2) time | O(n) space
7     public static List<Integer> threeNumberSum(int[] array, int targetSum) {
8         Arrays.sort(array);
9         List<Integer> triplets = new ArrayList<Integer>();
10        for (int i = 0; i < array.length - 2; i++) {
11            int left = i + 1;
12            int right = array.length - 1;
13            while (left < right) {
14                int currentSum = array[i] + array[left] + array[right];
15                if (currentSum == targetSum) {
16                    Integer[] newTriplet = {array[i], array[left], array[right]};
17                    triplets.add(newTriplet);
18                    left++;
19                    right--;
20                } else if (currentSum < targetSum) {
21                    left++;
22                } else if (currentSum > targetSum) {
23                    right--;
24                }
25            }
26        }
27        return triplets;
28    }
29 }
30
```

Solution 1   Solution 2   Solution 3

```
1 import java.util.*;
2
3 class Program {
4     public static List<Integer> threeNumberSum(int[] array, int targetSum) {
5         // Write your code here.
6         return null;
7     }
8 }
9
```

**Run or submit code when you're ready.**