

```

1  /*Author: Bochen (mddboc@foxmail.com)
2  Last Modified: Tue Apr 10 22:28:44 CST 2018*/
3
4  /*Given an array S of n integers, are there elements a, b, c in S such that a + b +
5  c = 0? Find all unique triplets in the array which gives the sum of zero.
6
7      Note: The solution set must not contain duplicate triplets.
8
9      For example, given array S = [-1, 0, 1, 2, -1, -4],
10
11      A solution set is:
12      [
13      [-1, 0, 1],
14      [-1, -1, 2]
15      ]*/
16
17  import java.lang.System;
18  import java.util.*;
19  import java.lang.Math;
20  import java.util.HashMap;
21
22
23  class ListNode {
24      int val;
25      ListNode next;
26
27      ListNode(int x) {
28          val = x;
29      }
30  }
31
32
33  public class Main {
34      public static void main(String[] args) {
35
36          int[] nums = {-1,0,1,2,-1,-4};
37
38          Solution solution = new Solution();
39
40          List<List<Integer>> receive = solution.threeSum(nums);
41
42
43          System.out.println("haha");
44
45      }
46
47
48  }
49
50
51  class Solution {
52      public List<List<Integer>> threeSum(int[] nums) {
53
54          List<List<Integer>> returnValue = new ArrayList<>();
55
56          if (nums == null || nums.length < 3) {
57              return returnValue;
58          }
59
60          Arrays.sort(nums);
61          int numsLength = nums.length;
62          int maxPositiveValue = nums[numsLength - 1] + nums[numsLength - 2];
63          int startPointer = 0, endPointer = 0;
64          int sum = 0;
65          for (int i = 0; i < numsLength; i++) {
66
67              if (nums[i] > 0) {
68                  break;
69              }
70
71              if (i > 0 && nums[i] == nums[i - 1]) {
72                  continue;

```

```

73     }
74
75     if (nums[i] < -maxPositiveValue) {
76         continue;
77     }
78
79     startPoint = i + 1;
80     endPoint = numsLength - 1;
81
82     while (startPoint < endPoint) {
83         if (startPoint != i + 1 && nums[startPoint] == nums[startPoint
84             - 1]) {
85             startPoint++;
86             continue;
87         }
88         if (endPoint != numsLength - 1 && nums[endPoint] ==
89             nums[endPoint + 1]) {
90             endPoint--;
91             continue;
92         }
93
94         sum = nums[i] + nums[startPoint] + nums[endPoint];
95         if (sum > 0) {
96             endPoint--;
97         } else if (sum < 0) {
98             startPoint++;
99         } else {
100             returnValue.add(Arrays.asList(nums[i], nums[startPoint], nums[endP
101                 ointer]));
102             startPoint++;
103             endPoint--;
104         }
105     }
106
107     return returnValue;
108 }

```