题目：

Given an array *S* of *n* integers, are there elements *a*, *b*, *c* in *S* such that *a* + *b* + *c* = 0? Find all unique triplets in the array which gives the sum of zero.

**Note:** The solution set must not contain duplicate triplets.

For example, given array S = [-1, 0, 1, 2, -1, -4],

A solution set is:

[

[-1, 0, 1],

[-1, -1, 2]

]

给一个sum值，找出数组中哪三个值相加等于sum，不能重复。

从这个题中我学会了什么？

（1）public static List<List<Integer>> threeSum(int[] nums){}

当给出这样的返回值的时候，不能直接new ArrayList<LinkedList<Integer>>

必须要分开新建

List< List<Integer>> reslist = new ArrayList< >();

List<Integer> res = new LinkedList<>();

然后用reslist.add(res);合并

（2）判断List的大小用List.size()并不是length()

只有String、stringbuffer、stringbulider是length()

数组是 nums[].length，没有括号

（3）清空list

1.将原有list 从新new一次

List<Integer> relist = new LinkedList<>(); //这样做原来的数据没有释放很占内存

2.调用clear()函数

relist.clear();

3.一个元素一个元素的remove

4.直接将list赋值null

relist=null；//但是再次打印relist的时候会打印出null的字符串

5.用removeAll()方法，参数添要删除的list，在这填写自己的list

relist.removeAll(relist);

（4）判断list是否为空

1.list.size()==0 ?

2.list.isEmpty==true ? false

回归题目，没啥可聊的。见代码吧。