**3Sum**

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.

class Solution **{**

public List**<**List**<**Integer**>>** threeSum**(**int**[]** nums**)** **{**

Arrays**.**sort**(**nums**);**

List**<**List**<**Integer**>>**list **=** **new** LinkedList**<>();**

int bot**,**top**,**sum**;**

**for(**int i**=**0**;**i**<**nums**.**length**-**2**;**i**++){**

**if(**i**==**0 **||** **(**nums**[**i**-**1**]!=**nums**[**i**])){**

bot**=**i**+**1**;** top**=**nums**.**length**-**1**;** sum**=**0**-**nums**[**i**];** //Sum represents the head the you are taking

**while(**bot**<**top**){**

**if(**sum **==** nums**[**bot**]+**nums**[**top**]){**

list**.**add**(**Arrays**.**asList**(**nums**[**i**],** nums**[**bot**],** nums**[**top**]));**

**while(**bot**<**nums**.**length**-**1**&&**nums**[**bot**+**1**]==**nums**[**bot**])**bot**++;**

**while(**top**>**0**&&**nums**[**top**-**1**]==**nums**[**top**])**top**--;**

bot**++;**

top**--;**

**}else** **if(**nums**[**bot**]+**nums**[**top**]<**sum**)**bot**++;** //You dont need to check all tha cases

**else** top**--;**

**}**

**}**

**}**

**return** list**;**

**}**

**}**

**-Ordering the arrays is the first good idea for not getting duplicates.**

**-You need to do Bidirectional search (have a top and a bottom index) for making a faster algorithm.**