Arrays - 3

In This Lecture

3 Sum Problem

3 Sum Problem

Given an integer array nums, return all the triplets such that nums[i] + nums[j] + nums[k] == 0 for distinct i,j and k

Notice that the solution set must not contain duplicate triplets.

Input: nums = [-1,0,1,2,-1,-4]

Output: [[-1,-1,2],[-1,0,1]]

^ ^

Brute force $\longrightarrow$ $O(n^3)$

3 Sum Problem

2 Sum problem ⟶ K

[2 1 -3 4 0 3]

```
① for(i=0 ⟶ n){
      for(j=i+1 ⟶ n){

      }
  }
```

$\boxed{O(n^2)}$

② Sort the array

3 Sum Problem

2 Sum

$a = [-3, 0, 1, 2, 3, 4]$ // $O(n \log n)$

$k = 3$ $l \rightarrow$ $r \leftarrow$

$+$

$O(n)$

$\rightarrow \boxed{O(n \log n)}$

$\rightarrow \cancel{1} \rightarrow \cancel{4} \rightarrow 3$

3 Sum

$$a[i] + a[j] = (-k) + (k)$$

$$+\, a[m] = 0$$

$a[m]$

$$a[i] + a[j] = -a[m] \rightarrow \text{2 Sum}$$

$$a[i] + a[j] + a[m] = 0$$

3 Sum Problem

a = {4, 2, -1, -3, 0, 1, 2, 3}; l

r

[-3, -1, 0, 1, 2, 2, 3, 3, 4]

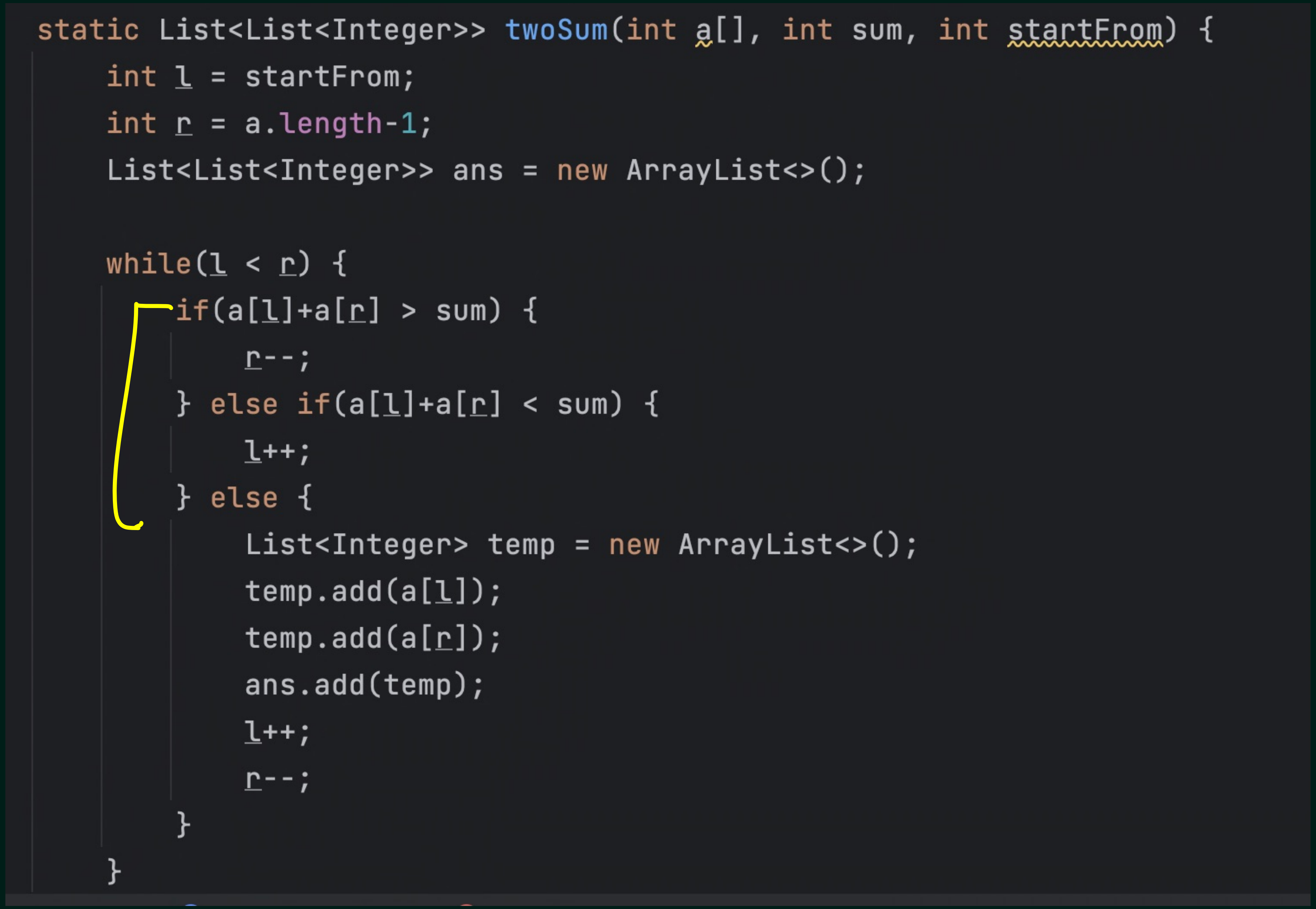
sum = 5

Set

[[1, 4]

[2, 3]

]



```java
static List<List<Integer>> twoSum(int a[], int sum, int startFrom) {
    int l = startFrom;
    int r = a.length-1;
    List<List<Integer>> ans = new ArrayList<>();

    while(l < r) {
        if(a[l]+a[r] > sum) {
            r--;
        } else if(a[l]+a[r] < sum) {
            l++;
        } else {
            List<Integer> temp = new ArrayList<>();
            temp.add(a[l]);
            temp.add(a[r]);
            ans.add(temp);
            l++;
            r--;
        }
    }
```