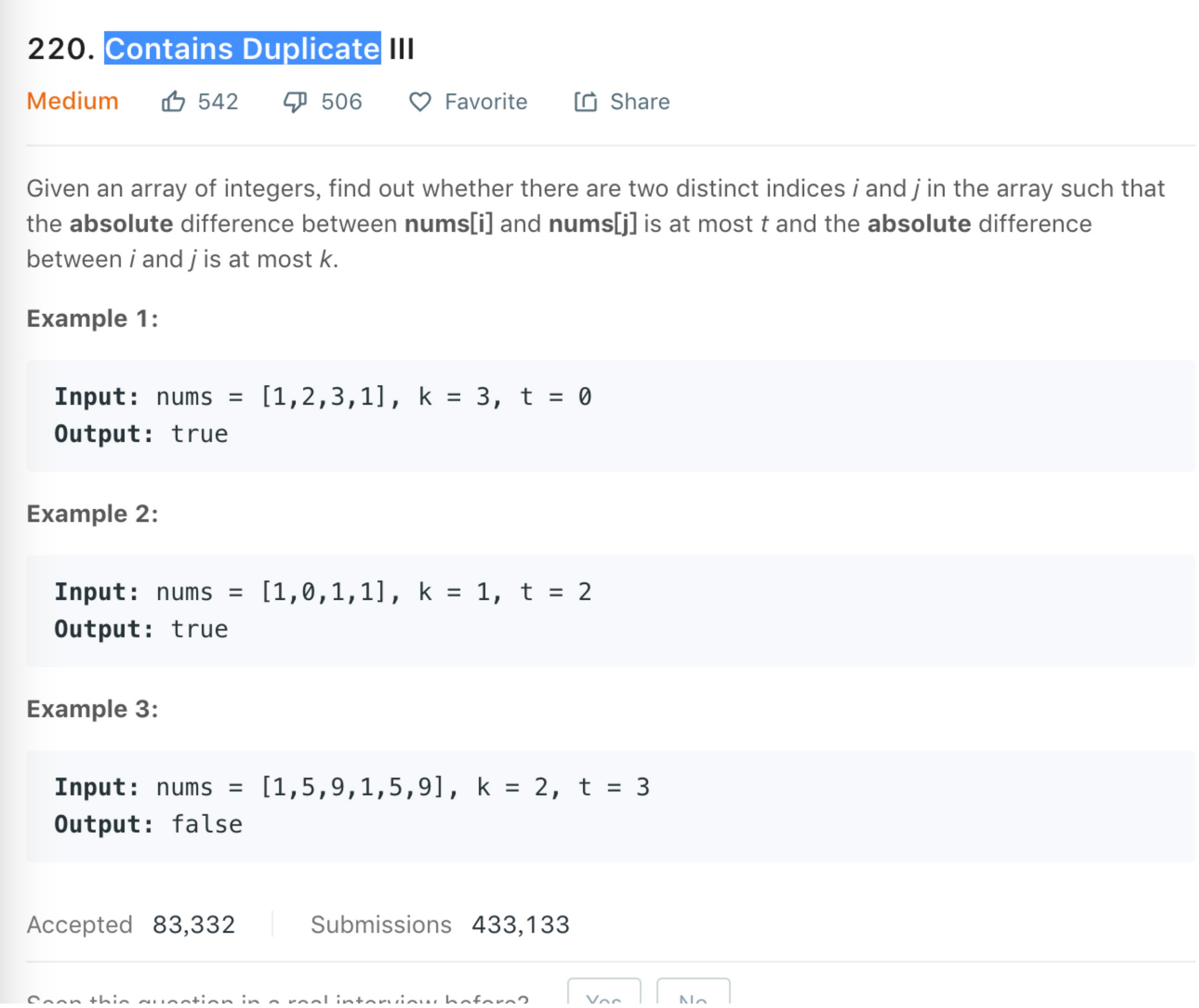


220. Contains Duplicate III

Given an array of integers, find out whether there are two distinct indices *i* and *j* in the array such that the **absolute** difference between **nums[i]** and **nums[j]** is at most *t* and the **absolute** difference between *i* and *j* is at most *k*.

Example 1:

```
Input: nums = [1,2,3,1], k = 3, t = 0
Output: true
```

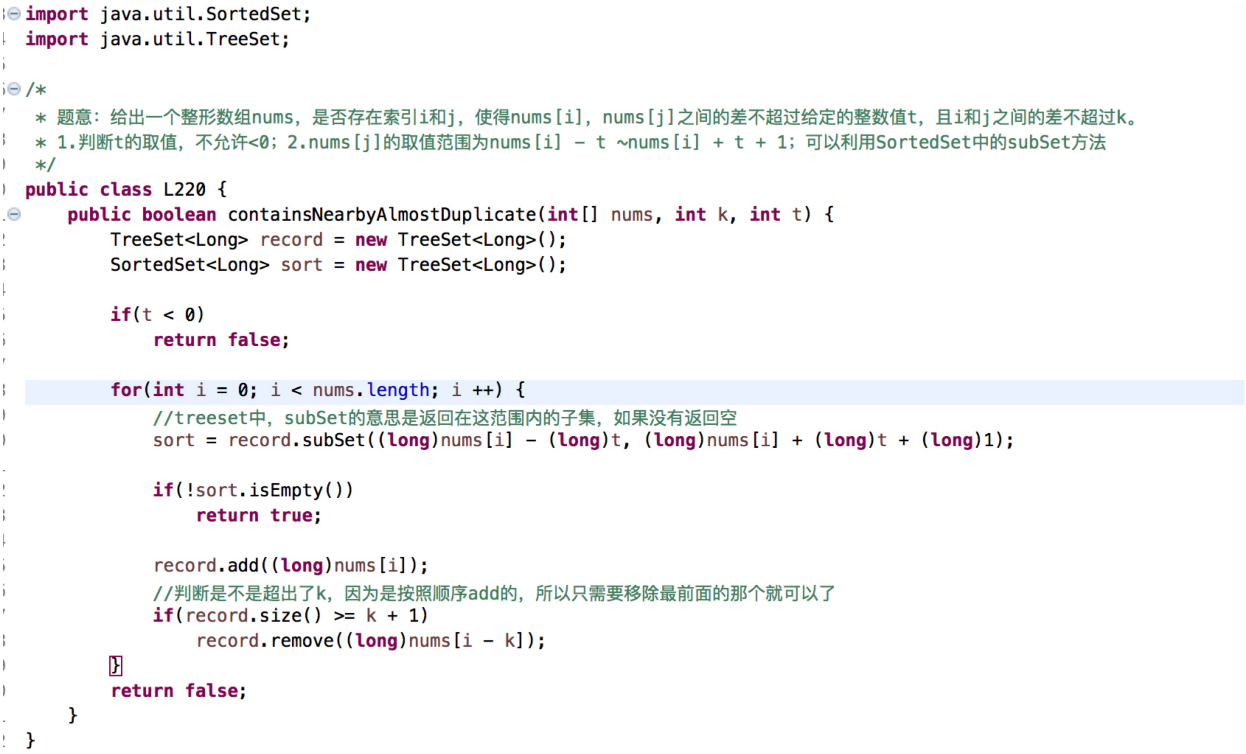
Example 2:

```
Input: nums = [1,0,1,1], k = 1, t = 2
Output: true
```

Example 3:

```
Input: nums = [1,5,9,1,5,9], k = 2, t = 3
Output: false
```

Accepted 83,332 | Submissions 433,133



```java
import java.util.SortedSet;
import java.util.TreeSet;

/*
 * 题意：给出一个整形数组nums，是否存在索引i和j，使得nums[i]，nums[j]之间的差不超过给定的整数值t，且i和j之间的差不超过k。
 * 1.判断t的取值，不允许<0；2.nums[j]的取值范围为nums[i] - t ~nums[i] + t + 1；可以利用SortedSet中的subSet方法
 */
public class L220 {
    public boolean containsNearbyAlmostDuplicate(int[] nums, int k, int t) {
        TreeSet<Long> record = new TreeSet<Long>();
        SortedSet<Long> sort = new TreeSet<Long>();

        if(t < 0)
            return false;

        for(int i = 0; i < nums.length; i ++) {
            //treeset中，subSet的意思是返回在这范围内的子集，如果没有返回空
            sort = record.subSet((long)nums[i] - (long)t, (long)nums[i] + (long)t + (long)1);

            if(!sort.isEmpty())
                return true;

            record.add((long)nums[i]);
            //判断是不是超出了k，因为是按照顺序add的，所以只需要移除最前面的那个就可以了
            if(record.size() >= k + 1)
                record.remove((long)nums[i - k]);
        }
        return false;
    }
}
```