HashTable 哈希表

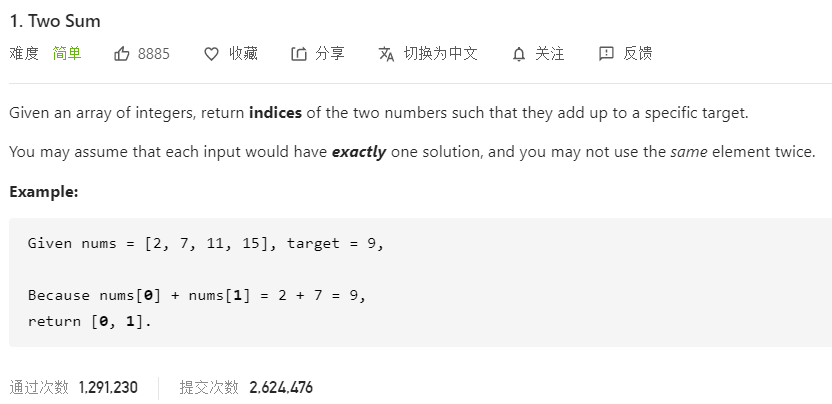
HashMap的put 如果key一样则会把value替换成新的value

put进会从小到大排 比如put 1，3，2 则最后是1，2，3

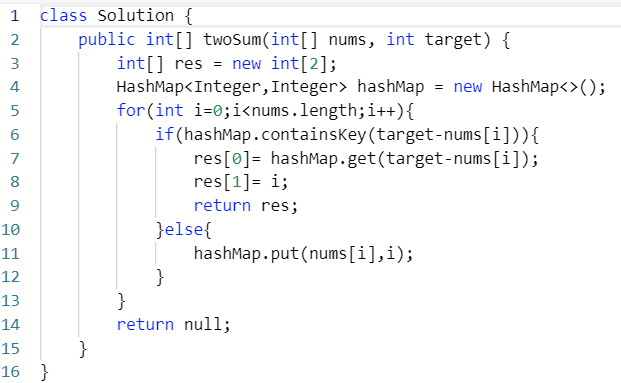
get会返回对应的key的value

HashMap不同于arrayList，arrayList是有序的，每个元素都有对应的index，会耗时间，而hashMap是根据查找key就可以找到对应的key和value

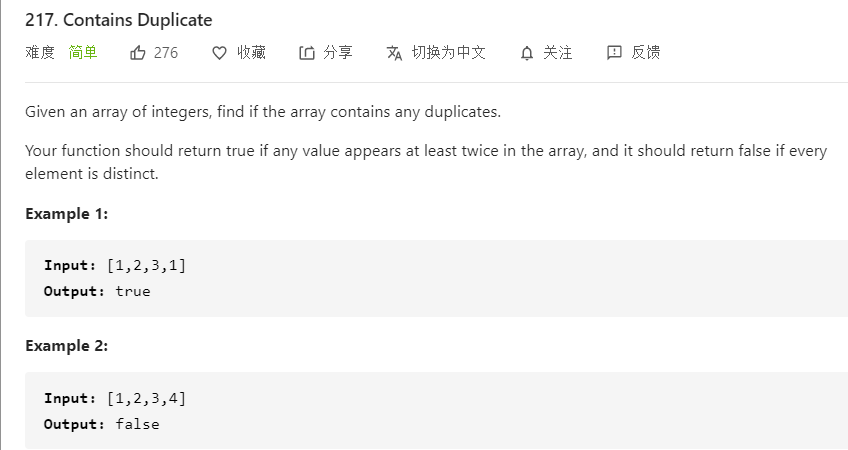
1.



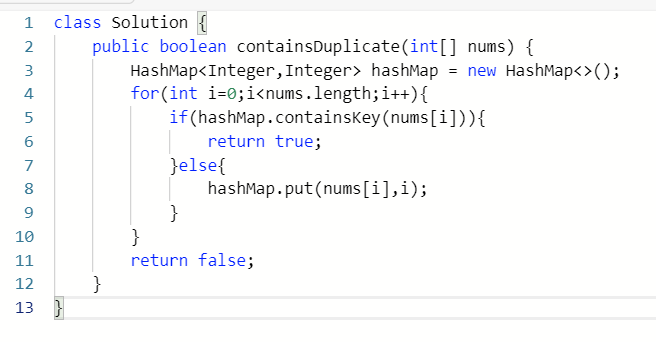
解法：用HashMap，先看有没有HashMap上有没有target - 他的key 没有的话就put到HashMap上 containsKey是精髓，数组不能直接用contains检查有没有与target差的那个值--要把array变成List（asList）



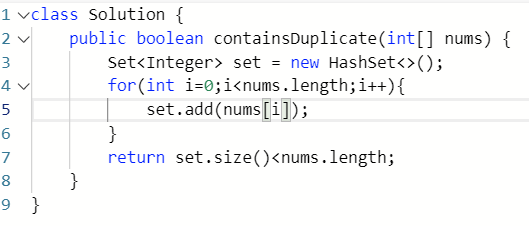
2.



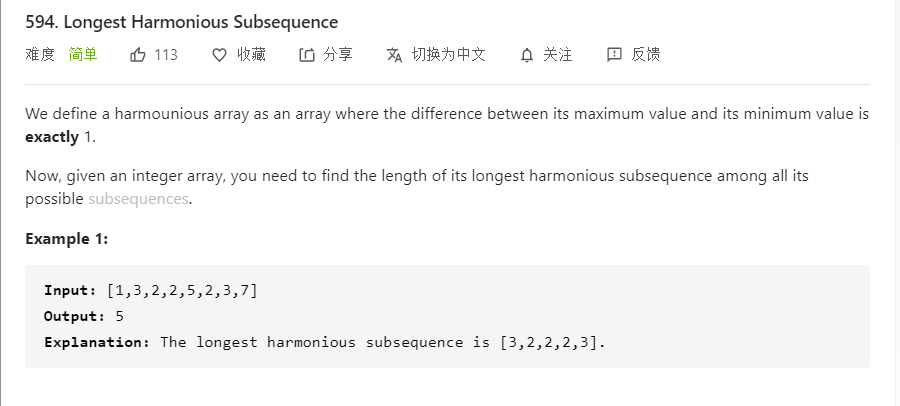
解法1：用hashmap判断有没有containsKey



解法2：直接用set去加，这样就不会加重复的，然后比较set的size是不是比给的小



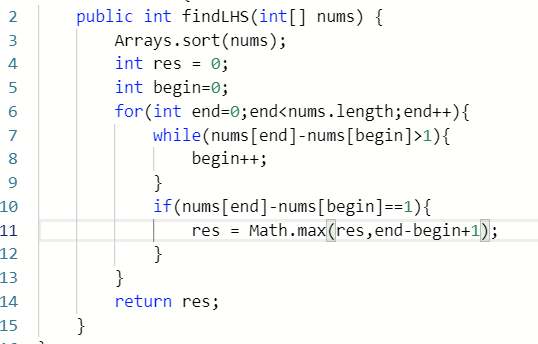
3.



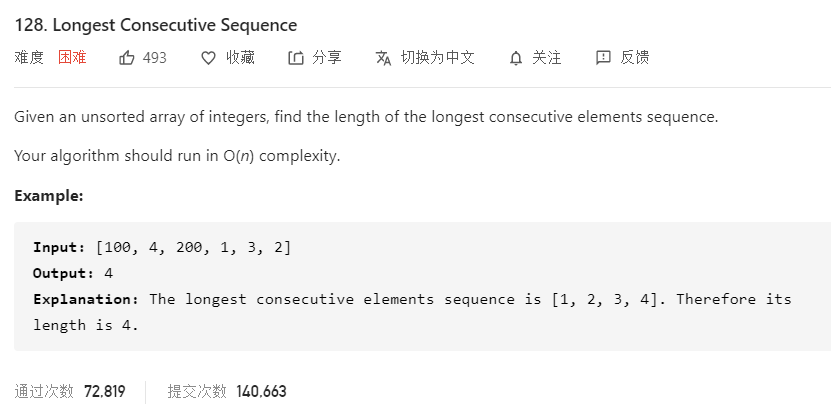
解法1：用hashMap，key来代表每个nums里的元素，用value来表示这个数在nums里出现的次数，之后用一个变量来记录每个数和比他大1的数出现的总次数(但前提是这个数要有比他大1的数)再看哪个数的的自己+大1的出现次数最大



解法2：用数组，先把数组用Arrays.sort()排序好，再用两个指针begin和end，如果end-begin=1，则返回的值取原来和现在end-begin+1的最大值，如果end-begin>1则把begin往后移一位，否则end++



4.



解法1：用hashMap，对于给的数组里的每个数如果不在hashmap里进行取key小1的value和key大1的value,（没有大1或小1就用0来代替）然后加起来得到，得到总长度，并且判断是不是最长，用res来记录最长的长度，再把这个数放进去with value -1，最后把连续的左右两个端点的数字给上value=length作为key放到hashMap中

当给的数组的长度大于1的时候，返回值至少为1

其实就是看比自己大1和小1的key的value，把他们加起来看看是不是最大的

每个loop先得到自己这个值小1和大1的key的value，因为返回的数组不能重复，所以hashMap里同样的key只加一次



解法2：用HashSet，先加到set里， 这样就没有重复的元素了，再对于每个元素进行查找比这个数大1或小1的数，如果存在一个就+1，最后把结果加起来(通过一个个remove)

