1. 力扣146LRU缓存机制

用LinkedHashMap(可以按照顺序put – 放到末尾)

get里用.keySet().contains()检查key是否存在，得到value后把这个元素重新插到末尾（先remove 再put）没这个key直接return -1，

put里如果没满+没存在就插，存在就插到末尾(先remove再put)，如果不存在，而且满了，就用iterator(类型是map.Entry<Integer,Integer>) = map.entrySet().iterator()，然后往后移一个然后删掉



2. 三数之和 找到所有**不重复**且和为0的三元组 (a,b,c) a+b+c=0

**先排序** -> a <b <c

1.要和上一次枚举的值不同才枚举，比如0，1，2，2，第一次枚举了012，第二次可以跳过第二个2.从小到大枚举b 从大到小枚举c

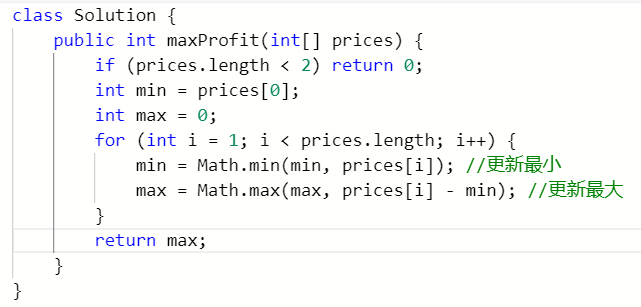
时间复杂度为O(N^2)，因为b从左到右移动一个位置，c从右往左移动若干个 + 外部的O(N)

If if while去重



3. 力扣121 买卖股票最佳时期

用min和max记录每次loop的最大和最小



4. K个一组翻转链表

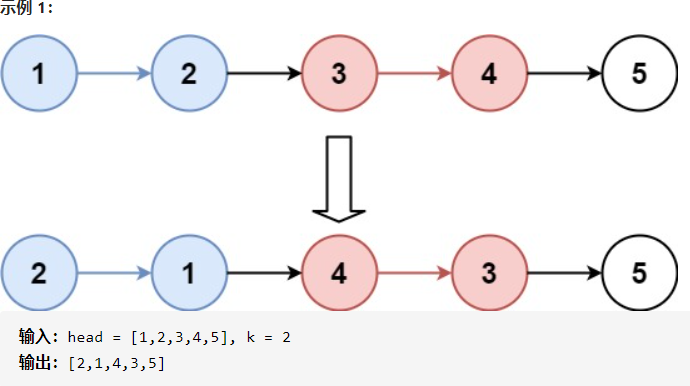
先创一个dummyNode，pre=dummy cur=head dummy.next = head

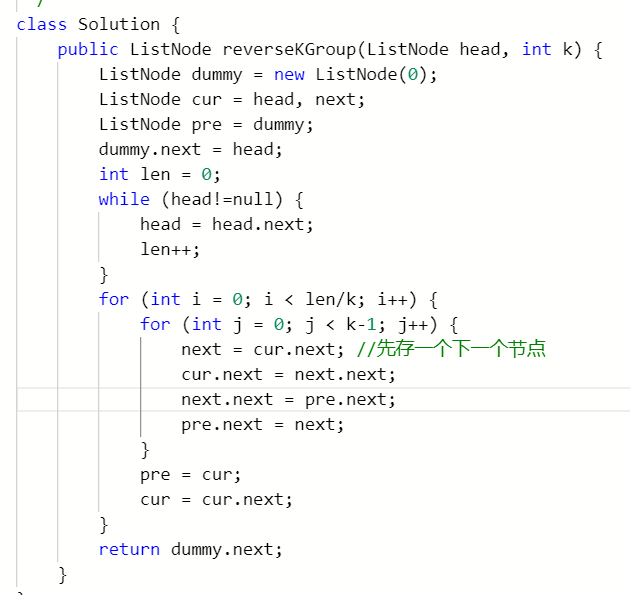
先得到链表的长度

对于每k个节点(len/k)，处理j<k 123 -> 213 -> 321

用dummyHead

先把当前指向下下个，把下个指向最左边的(也就是pre)，把pre指向下个。一个k段操作完再变pre和cur





#### 5. [215. 数组中的第K个最大元素](https://leetcode-cn.com/problems/kth-largest-element-in-an-array/)

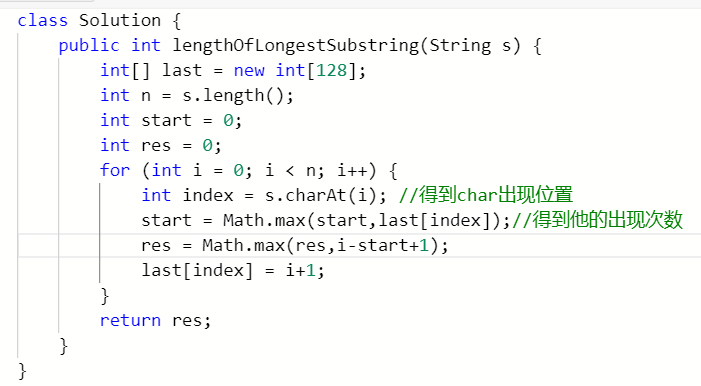
用二分找第k位左边比他小右边比他大

这个位置通过每次都partition，如果partition在k左边就去右边找

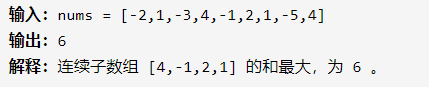
对于每次partition，找到左标比pivot大，右标比pivot小然后交换，这样左右标左右两边从小到大了再把pivot换过来，然后看pivot是不是正好是k



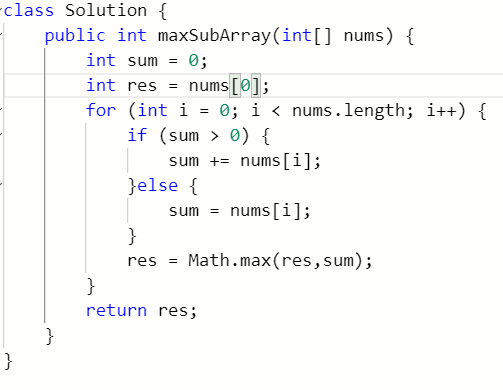
#### 6. [力扣第3题 无重复字符的最长子串](https://leetcode-cn.com/problems/longest-substring-without-repeating-characters/)



7，力扣53最大子序列和

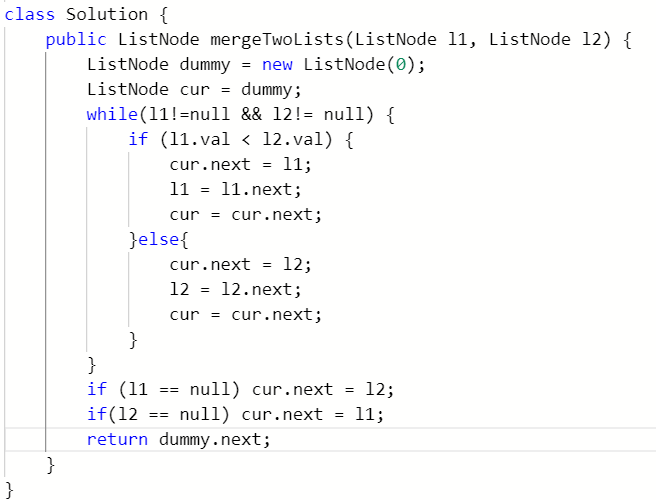


Loop里用sum去记录，如果sum<=0就没办法给后面做贡献了，所以直接等于当前数重新开始，如果>0就可以+当前数字继续，每次loop都要把res和sum比较取最大值

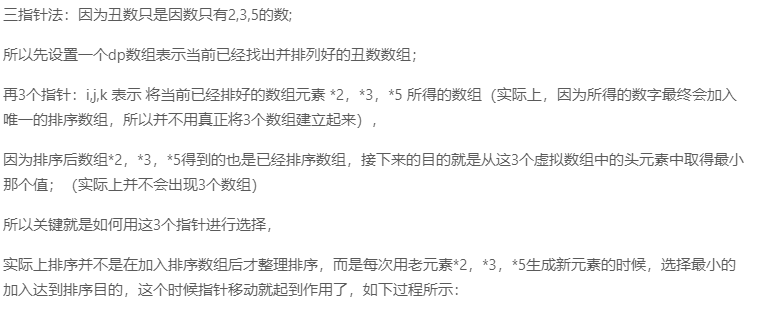


8. 合并有序列表 保证合并后的顺序

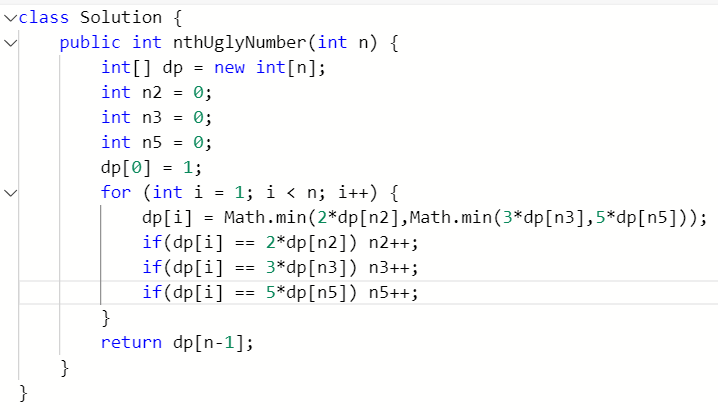
用dummyHead 从l1和l2分别取最小的放到dummy的后面，最后因为l1 l2长度可能不一样，所以l1没有直接把l2接在后面，反之接l1 即可



9.264丑数2



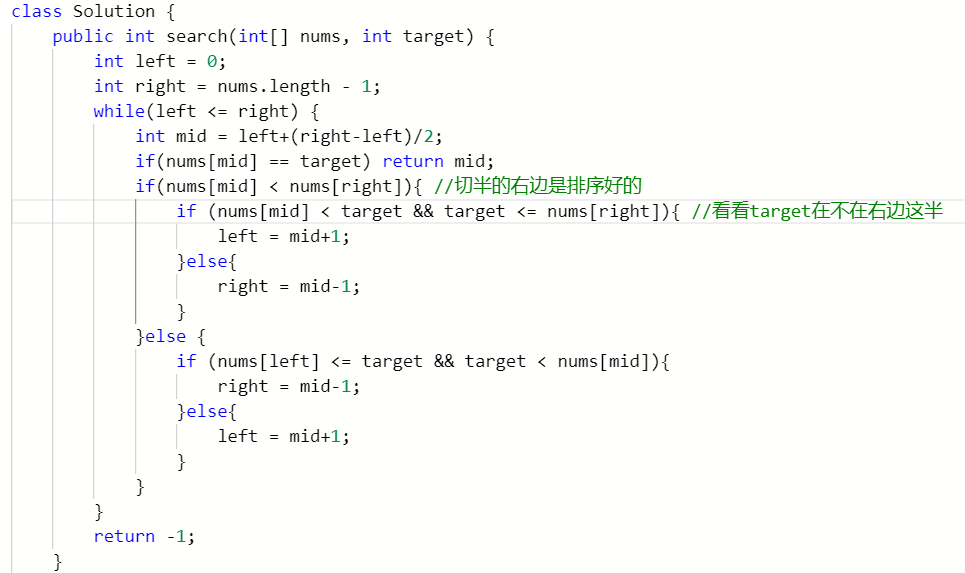
第一个位置变1，之后i..最小(2\*dp[n2] 3\*dp[n3]..) 先找哪个最小，最把选出来的n+1



10. 33搜索旋转数组 要求时间复杂度O(logn)

用二分的方法去找这个数组是不是有这个值

把数组一分为二，去有序的数组里找，如果中间数比最后边的小，则是有序的，反之去左半边，然后去确定target在不在这一范围内，在的话就移动left/right指针缩小范围，在left<=right的范围里找直到找到



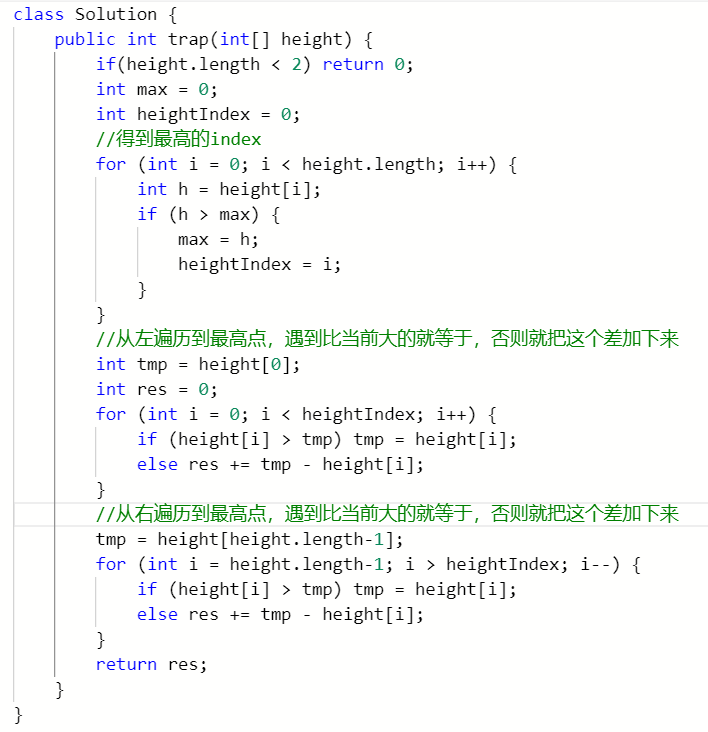
11. 102层序遍历 BFS

用queue，每次加入节点，然后节点自己加到返回的list把左右儿子加到queue，再把自己兄弟加到list，把兄弟的左右儿子加到queue，直到一层节点都遍历完(用一个size去记录这层节点数量，每轮只放一层，也就是知道size=0，也就是放size个元素从queue到list)



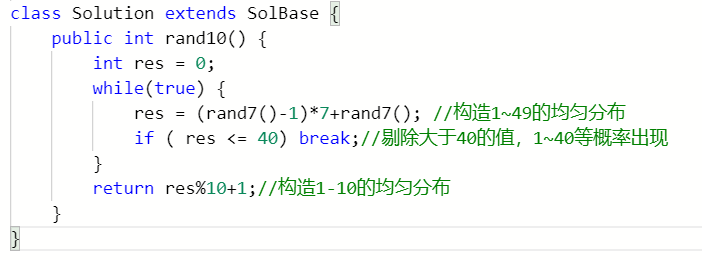
12. 接雨水 3个for loop 找最高点 从左到最高点 从右到最高点

遍历数组得到最大值的位置，分别从左到最高和从右到最高遍历，遇到比自己大的就等于，否则把差加到res



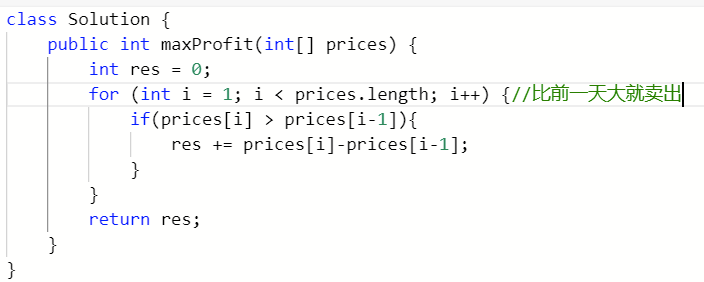
13. rand7实现rand10

构造1-49 去掉40-49，1-40等概率出现



14.股票买卖问题2

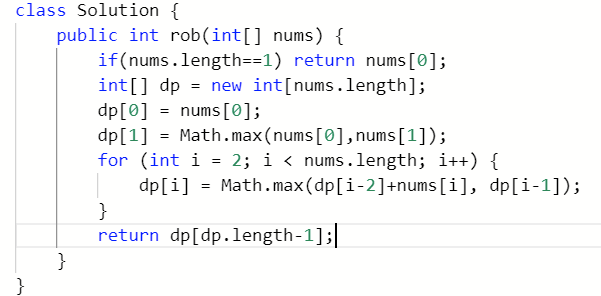
多次买卖股票达到最大效益



15.打家劫舍

动态规划 i位置代表i之前的最大收益

先**填满前2个位置**，然后遍历每次取MAX(前一次, 前前一次+这次)



16.169多数元素 找数组里出现次数超过一半的数字

解法：从第一个数开始count=1，遇到相同的就加1，遇到不同的就减1，减到0就重新换个数开始计数，总能找到最多的那个

因为个数超过总数一半去和别人一对一消耗，最后肯定是剩下这个超过一半的数