April 1, 2020. Wednesday

**LC 136 single number**
Concept

- If we take XOR of zero and some bit, it will return that bit
  - $a \oplus 0 = a$
- If we take XOR of two same bits, it will return 0
  - $a \oplus a = 0$
- $a \oplus b \oplus a = (a \oplus a) \oplus b = 0 \oplus b = b$

So we can XOR all bits together to find the unique number.

In java , XOR operator is " ^ "

## Binary search start here from HuaHua

**LC 35 search insert position (google like this )**

Binary search is a search algorithm that find the position of a target value within a sorted array.

In jave :

Integer **Overflow**

Let us now stress the fact that pivot = (left + right) // 2 works fine for Python3, which has arbitrary precision integers, but it could cause some issues in Java and C++.

If left + right is greater than the maximum int value 2^{31} - 1, it overflows to a negative value. In Java, it would trigger an exception of ArrayIndexOutOfBoundsException, and in C++ it causes an illegal write, which leads to memory corruption and unpredictable results.

Here is a simple way to fix it:

**pivot = left + (right - left) / 2;  or  pivot = (right + left) >>> 1; (this use memory)**

**LC 278 first bad version  (binary search )**

Left < right 这里可以是 <  或者<=,  所以可以在好好思考一下

**LC 34. Find First and Last Position of Element in Sorted Array  (good question )**

Initial a integer array with value ： **int res[] = { -1, -1 }**

**LC 202 happy number:**

用hashSet简单一点

如何得到十位上的数字，个位上的数字，等等

**April 3 Friday:**

**LC 53 maximum subarray**

Dp

**LC 981. Time-Based Key-Value Store**

Great question, need to review

**April 4 Saturday:**

**LC 283 move zeros**

- Different between subsequence and subarray, substring=====>>
  - Subsequence: no need to be consecutive
  - Substring: need to be consecutive
  - Subarray is in array, not in string: max stock value

**LC  1404 Minimum Subsequence in Non-Increasing Order**
Binary  representation,  decimal representation
A binary number ,from right to left, if the string ends in '0' then the number is even otherwise it is odd.

## LC 1405. Longest Happy String

**LC  546. Remove Boxes**
**https://leetcode.com/problems/remove-boxes/discuss/101310/Java-top-down-and-bottom-up-DP-solutions**

**Burst Balloons, zuma game, remove box, strange printer**

**April 6**
**LC 49. Group Anagrams**
**An anagram is string consist of same character but not same order**

**LC 81 Search in Rotated Sorted Array II**
**Has duplicate**
As follow up of LC 33 Search in Rotated Sorted Array
Problem is how to deal with duplicate

**LC33. Search in Rotated Sorted Array**
**两种情况来考虑**
345612
451234

**No duplicate**

- `.length` is a field, containing the capacity (NOT the number of elements the array contains at the moment) of arrays.
- `length()` is a method used by Strings (amongst others), it returns the number of chars in the String; with Strings, capacity and number of containing elements (chars) have the same value.
- `size()` is a method implemented by all members of Collection (lists, sets, stacks,...). It returns the number of elements (NOT the capacity; some collections even don´t have a defined capacity) the collection contains.

Array.length
string.length() , length is constant
Size is for mutable
这里好像有点问题， 因为List<Boolean> hasApple 这个也是用 size(), 好像之前理解的的不太对

**Binary Search huahua video, youtube**

Template : https://www.youtube.com/watch?v=v57INF2mb_s
[ L , R) 左闭右开

```
def binary_search( l, r):
    while l < r :
        m = l + ( r - l ) // 2
```

```
            if  f ( m ) :   return  m                    #  optional
            if   g (m):
                 r  = m                                  # new range [l,   m)
            else:
                 l = m + one                             # new range  [m+1, r)
        return  l                                        # or not found
#  分成了三部分,        m左边，m 和 m右边
```

Example 2 :  return the lower_bound /  upper_bound  of a val in a sorted array
lower_bound(x):  first index of i, such that A[i] >= x
upper_bound(x):  first index of i, such that A[i] >  x

**LC 69  sqrt (x)  边界问题**
**Upper bound and lower bound problem**

**Todo**
**LC875. Koko Eating Bananas**
**LC 378. Kth Smallest Element in a Sorted Matrix**

**Microsoft mock interview :**
**Lc 402. Remove K Digits**
https://leetcode.com/articles/remove-k-digits/

**LC 75 sort colors**
**My method is to use hashmap to count number of 0, 1, 2.**
**In-place one pass is,  if cur  meet 0, change cur with left,  if 1, continue,  if 2, change cur with right**

Follow up:

- A rather straight forward solution is a two-pass algorithm using counting sort.
  First, iterate the array counting number of 0's, 1's, and 2's, then overwrite array with total number of 0's, then 1's and followed by 2's.
- Could you come up with a one-pass algorithm using only constant space?

Read later
https://www.geeksforgeeks.org/java-util-linkedlist-peek-peekfirst-peeklast-java/?ref=rp

https://introcs.cs.princeton.edu/java/43stack/

April 8 Wednesday

**LC 876. Middle of the Linked List**
Whether the linked list is a loop or not

slow = fast = head
slow  = slow .next
fast = fast.next.next
这是返回 second one in the middle two,  如果是要返回first one,  need to change while condition from (fast != null && fast.next != null) to (fast .next.next != null && fast.next != null)  这样可以提前判断然后跳出 while

**LC 153 find the minimum value in a rotated sorted array  (middle)**
左闭右开 模版！！！ 花花的
Only need to compare with end value

**LC 154. Find Minimum in Rotated Sorted Array II (hard)**

Would it allow duplicates to affect the run-time complexity? How and why?

Only compare with the end,  if same,  end--

**LC 162  find peek element**

Your solution should be in **logarithmic** complexity.  ⇒ binary search

Need to explain why  binary search is work here

Recursive and iterative

**LC 852. Peak Index in a Mountain Array**

Find peak, compare condition is between  nums[i] and nums[i+1]

**LC 74. Search a 2D Matrix**

Convert the matrix into a sorted array based on the corresponding index in the array can get the row and col in 2d matrix

这里很有可能一开始start == end ,  因为matrix 是  [[1]],  start = 0, end = 1*1-1 = 0,

**Mock interview**

Nim game

**LC 461. Hamming Distance**

**bit operation**

**LC 836. Rectangle Overlap**

**LC 763. Partition Labels  (greedy )**

S.charAt(i) - 'a'　　java用这种方式把小写字母映射到1到26个数字上 (0 ~ 25 in array )

**Thursday, April 9**

**LC 844. Backspace String Compare**

Compare two string use: **str1. equals (str2)**

Get the value of a stack and into a string : **String.valueOf(ans)**;

Create stack: **Stack<Character> and = new Stack();**

Loop character in string: **for(char c: S.toCharArray())**

**LC 875　KoKo  eat bananas**

**LC 1011. Capacity To Ship Packages Within D Days**

Left bound is max(capacity) and right bound is sum(capacities )

Do binary search between two and find a minimum way to solve within D days

Similar questions:

- Find the Smallest Divisor Given a Threshold
- Divide Chocolate
- Capacity To Ship Packages In N Days
- Koko Eating Bananas
- Minimize Max Distance to Gas Station
- Split Array Largest Sum

**FB mock interview**

**LC 270. Closest Binary Search Tree Value**

**LC 71 simplify path**

A lot of thing in String use, need to check later again

String[] comp = path.split("/"); ## path is a string, use this

**Friday, April 10**

**LC 4 Median of two sorted arrays (binary search, hard )**

**April 12 , Sunday 复习这一个礼拜的**

**LC 49 Group anagrams:**

**注意**

1. **这不是一个string， 而是list of string， 求长度是 length， 没有 “( )”**
2. **虽然返回值是 List<List<String>>， 可以返回map的value，放在新的 arrayList 里面**
3. **Map的key是： 把每一个string存在 char[] ca = new char[26] 里面，然后把这个变成string， 用 String strKey = String.valueOf(ca)，就可以根据string的组成分类，因为anagram的ca里面存的是一样的东西。**
4. **map.get(key).add(value)**
5. **if(!map.containsKey(strKey)) map.put(strKey, new ArrayList<>());**

**LC 81 Search in Rotated sorted array ||**

**Binary search 有重复数字**

1. **用模版也是可以的，即左闭右开，[start，end )， start = mid+1，end = mid**
2. **Mid = start + (end - start)/2 ， avoid overflow**
3. **分类的关键是能不能保证所以情况都考虑到**
    a. **If else 的点是：nums[start] < nums[mid] or contrast condition**
    b. **nums[start] < nums[mid]: 234501**
    c. **nums[start] > nums[mid]: 450123**

**LC 33 Search in Rotated Sorted Array**
**Binary , no duplicate**
这里分类的部分不是 or 而是 and， 思考为什么？？？
https://www.cnblogs.com/grandyang/p/4325648.html

**LC69 sqrt(x)**
注意的是value的数据类型
while(start <= end){
int mid = start + ( end - start)/2;
long num = **(long)** mid * mid;

**LC 875 KoKo eating Bananas**
**Range is [1，max(piles)]，做binary search 找到一个potential 来看能否吃完all piles**

**LC 378 k th smallest element in a sorted matrix**
**https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/discuss/85173/Share-my-thoughts-and-Clean-Java-Code**

TBC……...

**Monday , April 13**
**Contiguous Array (binary array, only has 1 and 0)**
**Two methods: 1, use another array to store the count of 1 and 0**
**2, use hashmap to store**

**LC 668 Kth smallest number in multiplication table (LC 378，786)**
Prime number 质数
难在数据规模太大， 用binary search

Tuesday nothing

**Wednesday, April 15**

Every day question: Product of Array Except Self

Follow up question is use O(1) space, which is great. Only create an answer array , use R to update the value calculated from R array.

## LC 719. Find K-th Smallest Pair Distance (hard)

Binary search + prefix sum 太难，不会

Binary search + dp 花花的视频

https://leetcode.com/problems/find-k-th-smallest-pair-distance/discuss/109082/Approach-the-problem-using-the-%22trial-and-error%22-algorithm

https://leetcode.com/problems/k-th-smallest-prime-fraction/discuss/115819/Summary-of-solutions-for-problems-%22reducible%22-to-LeetCode-378

## LC 786. K-th Smallest Prime Fraction (hard)

数字位数很重要，数据类型

https://leetcode.com/problems/k-th-smallest-prime-fraction/discuss/115819/Summary-of-solutions-for-problems-%22reducible%22-to-LeetCode-378

LC **501. Find Mode in Binary Search Tree**
**Traversal in treevit**

## LC 307. Range Sum Query - Mutable (important )

- Binary index tree： 解决的是前n项和的问题 当有许多update的时候，因为如果只是想知道前n项和，使用pre-sum也很好，但是不能解决有许多update的情况
- 构建 BIT， 使用二进制
- https://www.youtube.com/watch?v=CWDQJGaN1gY
    - it will be good if it is mentioned why 5 is expressed as 2^2 + 2^0 and not 2^0 + 2^2. The idea here is to go to the parent and then next x number of elements. I got confused when i saw the video. Then later i figured it out that it is expressed as parent index + some power of 2.
- https://www.cnblogs.com/xudong-bupt/p/3484080.html
- https://github.com/mission-peace/interview/blob/master/src/com/interview/tree/FenwickTree.java

Segment tree

## April 14
## LC 123. Best Time to Buy and Sell Stock III dp hard

## April 15 Friday
## LC 289. Game of Life
Scalability

## LC 272. Closest Binary Search Tree Value II
BST use inorder can has a increasing trend order

**April 16 Saturday**
LC 64 Minimum path sum
**滚动数组**的改法要看一下

**Weekly contest 185　　全是关于 string的问题**
**1.Reformat The String**
**基本功不行啊！！！**
StringBuilder　　看stringBuilder的好处是什么

LC 530. Minimum Absolute Difference in BST

- LC 783

Stack and queue ????????/

Inorder　　morris traversal.

## April 17 Sunday

**LC 701. Insert into a Binary Search Tree**
注意的是怎么create 一个新的node：　node.right = new TreeNode(val);

**LC 230. Kth Smallest Element in a BST**
最蠢办法，　inorder traversal得到sort序列
Follow up: What if the BST is modified (insert/delete operations) often and you need to find the kth smallest frequently? How would you optimize the kthSmallest routine?

- LC 146. LRU Cache

# LC 501. Find Mode in Binary Search Tree

Could you do that without using any extra space? (Assume that the implicit stack space incurred due to recursion does not count).

## April 20  Monday:

**LC 1008. Construct Binary Search Tree from Preorder Traversal**

Time complexity and space complexity

To do soon:

**LC 664. Strange Printer  (hard，dp)**

计数题目

Hint: Length of the given string will not exceed 100.  一般就是 N的3次方的题，如果是2次方应该是500～1000+

花花的视频

**LC 665. Non-decreasing Array**

https://leetcode.com/problems/non-decreasing-array/discuss/106849/C%2B%2B-Java-Clean-Code-6-liner-Without-Modifying-Input

**To do list:**

**LC 4. Median of Two Sorted Arrays  题不错**

**LC 33. Search in Rotated Sorted Array**

- 注意如果判断target ！= nums[mid]， 那之后可以不用判断 target >= 或者<= nums[mid]，不需要等于， 但是 target 与nums[start] and nums[end]还是有可能相等的

**LC 123. Best Time to Buy and Sell Stock III**

- **Solution is little complexity， check this out**
- **https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii/discuss/135704/Detail-explanation-of-DP-solution**

**Mock interview : String！！！**
**LC 824 goat latin:**
**String: use in `S.split("\\s+")` it will cover the case of consecutive spaces**

**LC166. Fraction to Recurring Decimal**
**处理string是个问题**

**April 21 Tuesday:**
**这一天的leetcode everyday challenge 的一个blog：**
Leftmost Column with at Least a One

**https://codeforces.com/blog/entry/62602n**

**http://www.cs.cornell.edu/courses/cs211/2006sp/Lectures/L06-Induction/binary_search.html**

**Check binary search correction could use 2 value ,  0, 1 then implement algorithm**

**April 22 Wednesday:**

preSum: use of a cumulative sum array, sum, such that sum[i] is used to store the cumulative sum of nums array upto the element corresponding to the (i - 1) index, thus , to determine the sum of elements for subarray nums[i : j],  use sum[j+1]= sum[i]

**LC 166. Fraction to Recurring Decimal**
Long division

**LC 99. Recover Binary Search Tree  (hard,  but great question)**

**Do not use stack, use Deque<TreeNode> stack = new ArrayDeque<>(); ??**
**https://docs.oracle.com/javase/8/docs/api/java/util/Stack.html**
**https://www.geeksforgeeks.org/arraydeque-in-java/**
**https://chengfeng96.com/blog/2018/01/20/Java%E4%B8%AD%E7%94%A8Deque%E6%8E%A5%E5%8F%A3%E4%BB%A3%E6%9B%BFStack%E6%8E%A5%E5%8F%A3%E5%AE%8C%E6%88%90%E6%A0%88%E5%8A%9F%E8%83%BD/**

Check :  Morris Inorder Traversal

April 26 Sunday:
Everyday:
LRU cache
双向链表能在O(1)时间里面添加和删除节点，单链表不行
Java也有双向链表 LinkedList， 但是 LinkedList封装的太深，没有能在O(1)时间内删除中间某个元素的API ((C++）的list有个splice()， O(1)时间可以处理，所以只能实现一个双向链表

## April 27 Monday:

LC 30 day challenging:
**Maximal square**
Rolling array to reduce space
Use a temp to store previous dp[j]

**LC 108. Convert Sorted Array to Binary Search Tree** **build bst**
Java random ：Random rand = new Random();

**LC 450. Delete Node in a BST**
How to delete a node in bst

- **How to find predecessor One step left and then always right**
- **How to find successor : One step right and then always left**

**下一步做做string的题，基础一点**

Repeat this week:
**LC 3. Longest Substring Without Repeating Characters** **(**sliding window, hashset, string**)**
**没太明白 (done)**

HashSet<> set = new HashSet<>();
set.add(),  set.remove()

## LC 15 3sum  (two pointers)

**Add new list to arrayList:  res.add(Arrays.asList(nums[i], nums[j], nums[k]));**

**if( i > 0 && nums[i] == nums[i-1]) continue;  这句括号里面的顺序一定不能变**

**LC  16. 3Sum Closest  (two pointer)**
Two pointer careful about the range of i , j, and k,  three pointers , especially i range in for loop,  from 0 to n-2

Mock interview fb
LC 496. Next Greater Element I
用hashmap的方法可以省去stack的使用，但是时间上来到了O(n to 2)

## Wednesday April 29, 2020

Every day LC Binary Tree Maximum Path Sum

LC 124  binary tree maximum path sum
LC 687 longestUnivaluePath  这几个都类似, 都可看花花视频
LC 543. Diameter of Binary Tree

Some shortest path algorithm
Floyd-Warshell algorithm
Get all the pairs of node to every other node shortest path

**Thursday  April 30 2020**

**LC every day:** Check If a String Is a Valid Sequence from Root to Leaves Path in a Binary Tree

String

无脑抄：
LC 30. Substring with Concatenation of All Words

May 1 Friday:
String
**LC 1119. Remove Vowels from a String**
StringBuilder use **append** to add new character

**LC 1165. Single-Row Keyboard**
Use simple int array to store the value character - 'a' to replace the use of hashmap

**1221. Split a String in Balanced Strings**

**LC 709. To Lower Case**
ASCII code for capital letter and lowercase letter

**LC 1180. Count Substrings with Only One Distinct Letter**
**Math**
**sum of arithmetic progression**

If a letter repeats n times, it forms n * (n + 1) / 2 valid substrings:

`"aaaa": "a" * 4, "aa" * 3, "aaa" * 2, and "aaaa" * 1 = 4 * (4 + 1) / 2 = 10.`

### LC 1309. Decrypt String from Alphabet to Integer Mapping

## String to Integer in Java – parseInt()

(char)(66)  will get letter 'B' ,

res.reverse().toString();  reverse a string
res.append((char)(97 + s.charAt(i) - '0' - 1));  得到i位置的数字对应的字母 （全小写）

## saturday May 2

**LC** Jewels and Stones
Hashset use add()

## Contest

1436. Destination City

for (List<String> l: paths) set.add(l.get(1));

## 1438. Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit

**Deque** 的用法，好好看看

## LC 1439. Find the Kth Smallest Sum of a Matrix With Sorted Rows

**Heap** ⇒ priority queue
https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html

Sunday May 3
LC 804 unique morse code words

### LC 1370. Increasing Decreasing String

using int[256] to store char instead of using int[26] with 'a' for string to store character, thus not only limited to lower case ,but also for upper case
注！！ while的花括号必须加，不然影响逻辑 {}

**1374. Generate a String With Characters That Have Odd Counts stupid question**

Arrays.fill(str, 'a');

## LC 657. Robot Return to Origin

Int[][] dir = new int[][]{{0,1},{0, -1}, {1,0},{ -1, 0}}

## LC 929. Unique Email Addresses

https://leetcode.com/problems/unique-email-addresses/discuss/186798/JavaPython-3-7-and-6-liners-with-comment-and-analysis.

**Java 里面string的单双引号问题， 如果不确定都用双引号～！！！！！！**

## LC 893 Groups of Special-Equivalent Strings

Arrays.toString()

**`Arrays.toString(new int[]{2,22})` gives you `[2, 22]` instead of `222`,**

## LC 824. Goat Latin

vowel (a, e, i, o, or u),
a consonant (i.e. not a vowel)

**做做 heap， priority queue**

## LC 703. Kth Largest Element in a Stream

pq default is min priority queue

## LC 451. Sort Characters By Frequency

```
List<Character> characters = new ArrayList<>(map.keySet());
        Collections.sort(characters, (a,b) -> map.get(b) - map.get(a));
```

https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html

**Review last week:**

LC 43. Multiply Strings

nums1.charAt(i) - '0'  use this to get the number in string

**1414. Find the Minimum Number of Fibonacci Numbers Whose Sum Is K**

**找个时间好好看看了。。。。。**

**1438. Longest Contiguous Subarray With Absolute Diff Less Than or Equal to Limit**

Use arrayDeque rather than linkedList

LC 239 , reference

https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit/discuss/609771/JavaC%2B%2BPython-Deques-O(N)

Java deque implementation:
http://tutorials.jenkov.com/java-collections/deque.html#java-deque-tutorial-video

1. **Generic types:**

```
Deque<String> deque = new ArrayDeque<>();
Deque<String> deque2 = new LinkedList<>();
```

2. // **Queue mode:**
   a.
```
deque.offerLast("Element 1");
deque.offerLast(" Element 2");

String element1 = deque.pollFirst();
String element2 = deque.pollFirst();
```

3. **// stack mode:**
   a.
```
deque.offerFirst("Element 3");
deque.offerFirst("Element 4");

String Element 3 = deque.pollFirst();
String Element 4 = deque.pollFirst();
```

## May 5, Tuesday

**LC 1180. Count Substrings with Only One Distinct Letter**

**LC 1165. Single-Row Keyboard**

**LC 929. Unique Email Addresses**
https://howtodoinjava.com/java/string/java-string-replaceall-example/

**LC 901. Online Stock Span**

Use monotone stack store  int[]{price,  currRes}
Deque<int[]> stack = new ArrayDeque<>();
https://leetcode.com/problems/online-stock-span/discuss/168311/C%2B%2BJavaPython-O(1) 有一些类似的题

## May 6 Wednesday

**LC 169. Majority Element**

**May 7 Thursday**

Everyday challenge

**May 8 Friday**

everyday

**LC 339. Nested List Weight Sum**

Three function:

1. isInteger()
2. getInteger()
3. getList()
4. addAll:  This method appends all of the elements in the specified collection to the end of this list, i

**LC 897. Increasing Order Search Tree**

Inorder traversal to get the val, then  add value to arraylist

**May 8 Saturday**

**Everyday**

**LC 559. Maximum Depth of N-ary Tree**

DFS & BFS both ok

**LC 104. Maximum Depth of Binary Tree**

和上面这道一样，dfs更快

**LC 872 leaf similar tree**

**LC 690. Employee Importance**
Careful about given IDE

**LC 733. Flood Fill**
**不能 newColor == cent color 不然endless loop**
If newColor is the same as color, after visiting a neighbour point of a starting point, The DFS function would visit the starting point again and this loop never stops.

**int[][] dir = new int[][]{{0,1},{0,-1},{1,0},{-1,0}}**

**LC 100. Same Tree**
Can use iteration or recursion,

**LC 257. Binary Tree Paths** **多做几次，有点问题**

**Sunday May 10**
**Everyday: LC** 997. **Find the Town Judge**

**LC 277. Find the Celebrity**
**类似997， 但是可以用greedy的方法**

**LC 111. Minimum Depth of Binary Tree**

**Saturday contest:**
**LC 1443 Minimum Time to Collect All Apples in a Tree** **huahua video**

**固定长度的list of list of integer可以这样创建：**
**List<Integer>[] graph = new List[n];**
**声明就是： List<Integer>[] graph**

**类似于：** `List<List<Integer>> graph = new ArrayList<>();`

## Mock interview Google

**LC 482. License Key Formatting**

String .toUpperCase()　和　toString()可以直接使用，不用一个一个的对应着用

**388. Longest Absolute File Path**

**May 11　　Monday:**
**Everyday:　flood fill**
**Still dfs/ bfs**
**LC 1302. Deepest Leaves Sum**
Just return last time the sum of node

M**ay 12　Tuesday**
**Everyday**
**LC　540. Single Element in a Sorted Array**

**1315. Sum of Nodes with Even-Valued Grandparent**

**366. Find Leaves of Binary Tree**
**往一个arrayList里面加一个List用add**
**Mock interview: check later**

## May 13 Wednesday:

**LC 402. Remove K Digits**

**Repeat last week:**
**LC** 59. Spiral Matrix II

**LC 104. Maximum Depth of Binary Tree**

- Recursive
- Iterative :　use queue to do　BFS

LC 111. Minimum Depth of Binary Tree

- Recursive
- Iterative:  use bfs more easier

**LC中国打卡：  single number：**

异或解法：异或运算满足交换律，a^b^a=a^a^b=b,因此ans相当于nums[0]^nums[1]^nums[2]^nums[3]^nums[4]..... 然后再根据交换律把相等的合并到一块儿进行异或（结果为0），然后再与只出现过一次的元素进行异或，这样最后的结果就是，只出现过一次的元素（0^任意值=任意值）

**May16 Saturday night:**

Set,  priorityqueue,  string,  HashMap entrySet()

**May 17 Sunday**:

Everyday:

Find All Anagrams in a String

https://leetcode.com/problems/find-all-anagrams-in-a-string/discuss/92007/Sliding-Window-algorithm-template-to-solve-all-the-Leetcode-substring-search-problem.  这个答案总结了好多用**Sliding Window algorithm template** to solve all the Leetcode substring search problem.

https://leetcode.com/problems/minimum-window-substring/discuss/26808/here-is-a-10-line-template-that-can-solve-most-substring-problems 10-line template that can solve most 'substring' problems

**Monday  May. 18**

**Everyday:**

**LC 438 . Find All Anagrams in a String**

Sliding window

**Tuesday May 19**

**LC Everyday**  Online Stock Span

**力扣的everyday：**1371. Find the Longest Substring Containing Vowels in Even Counts
可以查看中文lc里面人们的回答

Wednesday May 20

Kth Smallest Element in a BST, LRU cache, include “insert in BST”. “Delete in BSt ”

LC everyDay: 230. Kth Smallest Element in a BST

LC 25 reverse Nodes in K- group

关键在于怎么讲一个linkedList 反转：

Eg: dummy(p,c) -> 1 ->2 -> 3 变成 dummy() ->2 ->1 ->3:

就按照这个顺序连接就好。

1. *c = p.next;*
2. *p.next = c.next;*
3. *c.next = n.next;*
4. *n.next = c;*

- 类似： 206. Reverse Linked List
- 92. Reverse Linked List II　　todo soon

LC 124. Binary Tree Maximum Path Sum

May 21 Thursday:
LC everyday : Count Square Submatrices with All Ones (dp)
LC china everyday: 105. Construct Binary Tree from Preorder and Inorder Traversal

May 22, friday:
Feb. 29, 2020 451. Sort Characters By Frequency Average Rating: 5 string information
Everyday LC 451. Sort Characters By Frequency : how to use heap
**Bucket Sort**

**LC 459. Repeated Substring Pattern**
**Careful**

**LC 489. Robot Room Cleaner**

**May 23 Saturday**
**Contest 做了2道题**

## May 25 Monday
## LC 1035. Uncrossed Lines
**Dp， use rolling array**

**LC 286. Walls and Gates**
Bfs and dfs both ok
https://leetcode.com/problems/walls-and-gates/discuss/72748/Benchmarks-of-DFS-and-BFS

## LC 1008. Construct Binary Search Tree from Preorder Traversal

Insert node to a bst
Recursion and iteration both need to 会

**LC 303 and 307 range sum query - mutable and immutable**
花花视频
即使是immutable也要很小心
花花酱 Fenwick Tree / Binary Indexed Tree - 刷题找工作 SP3

Get lowbit(i): lowbit(x) = x & (-x)
在我的desktop的temp文件夹里面有代码

- LC 315. Count of Smaller Numbers After Self hard

https://visualgo.net/bn/fenwicktree
注意这棵树在update的时候要从1开始，虽然在fenwicktree里面的update没有从1开始，但使用时要从1，不然就卡在0出不去了

**Mock interview Google**

LC 788. Rotated Digits
LC 849. Maximize Distance to Closest Person （ two pointer ）

# May 27  Wednesday:

LC everyday :

## Possible Bipartition

https://www.w3schools.com/python/python_operators.asp

| Operator | Name | Description |
|---|---|---|
| & | AND | Sets each bit to 1 if both bits are 1 |
| \| | OR | Sets each bit to 1 if one of two bits is 1 |
| ^ | XOR | Sets each bit to 1 if only one of two bits is 1 |
| ~ | NOT | Inverts all the bits |
| << | Zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off |
| >> | Signed right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off |


因为这道题要染色， 在0和1之间转换，所以用XOR　^ 1 就可以保证在 0 和1 之间变换 ：　0 ^ 1 = 1，1 ^ 1 = 0

Create a global arrayList use :

- ArrayList<Integer>[] graph;
- graph = new ArrayList[N+1];

注意这道题给的note：

1. 1 <= N <= 2000
2. 0 <= dislikes.length <= 10000

边的数量小于N的平方, 说明这是一个比较sparse的图

**LC 785  is graph bipartition**：　一样的方法，图染色方法

**Microsoft mock interview**

- **Flipping an image**
- **Leaf-similar trees**

# May 28  Thursday:

**LC 338. Counting Bits**

**Observation:  if  x / 2 == y,  the number of set bits in x - the number of set bits in y <= 1**
**a number  (x >> 1) is equals  x / 2**

**May 29 2020 Friday**
**LC 207. Course Schedule  topological sort**
Check later

## June 1  2020,  Monday  DFS

**LC 366. Find Leaves of Binary Tree**
https://leetcode.com/problems/find-leaves-of-binary-tree/discuss/83778/10-lines-simple-Java-solution-using-recursion-with-explanation
https://www.cs.cmu.edu/~adamchik/15-121/lectures/Trees/trees.html

**LC 1448. Count Good Nodes in Binary Tree**

**June 4 Thursday**
**repeat**
**LC 25 reverse Nodes in k-group**

**June 4 Saturday**
**LC everyday:**
Queue Reconstruction by Height
Java override compare method

## Some ways to sort in Java

use  Arrays.sort(Comparator())
https://blog.csdn.net/tongyuehong/article/details/45318567

## Weekly contest 192

LC 1471. The k Strongest Values in an Array
1472. Design Browser History
1473. Paint House III

## June 7, 2020  sunday

LC 518. Coin Change 2

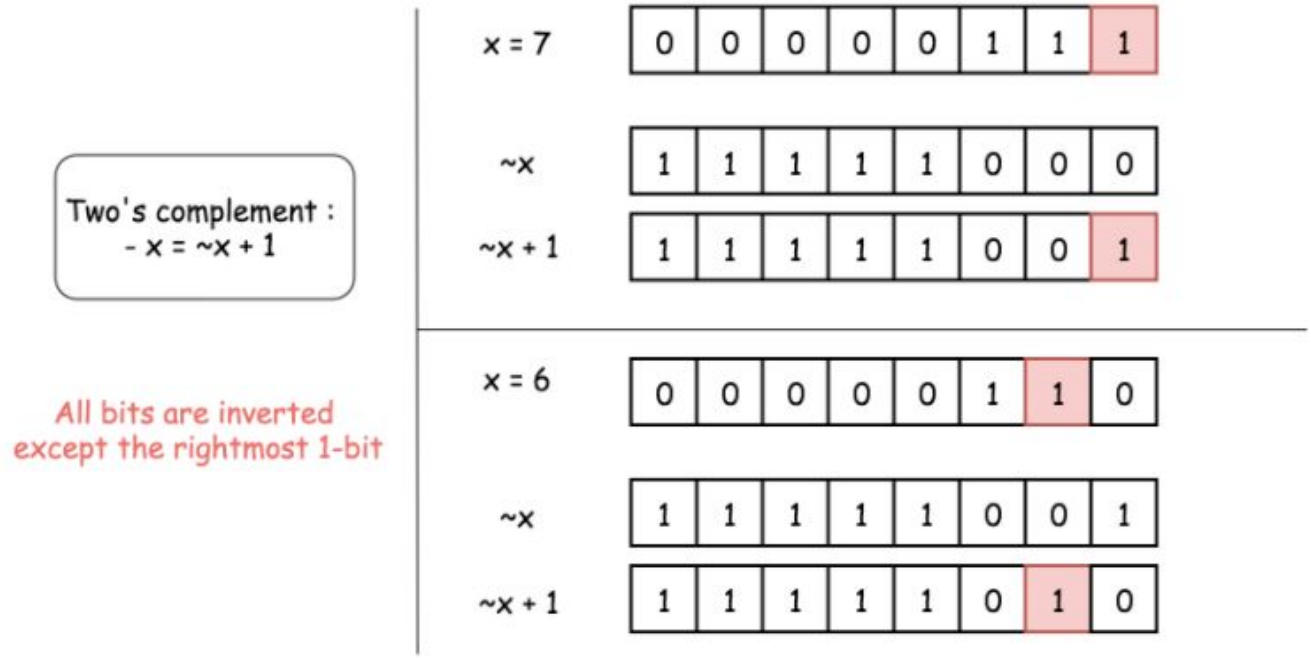
June 8 2020 Monday

LC 231. Power of Two

Bitwise  https://leetcode.com/problems/power-of-two/solution/

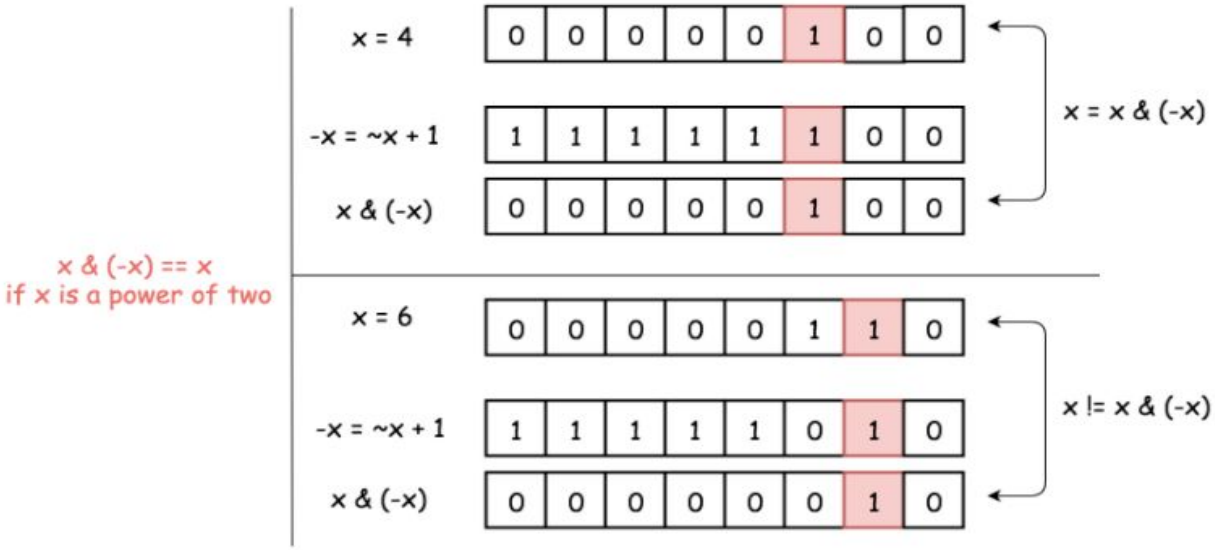
- How to get / isolate the rightmost 1-bit : `x & (-x)`.
- How to turn off (= set to 0) the rightmost 1-bit : `x & (x - 1)`.
- https://leetcode.com/articles/n-queens-ii/

In summary, $-x$ is the same as $\neg x + 1$. This operation reverts all bits of x except the rightmost 1-bit.




If this is power of 2 :

Hence a number is a power of two if `x & (-x) == x`.



LC 518 coin change 2
https://www.youtube.com/watch?v=jaNZ83Q3QGc
果然视频还是比文字好
**LC 322 coin change**

**June 9　Tuesday**
**Everyday LC**

Is subsequence: easy, but follow up is great, use  binary search solution

**June 13 Saturday**
**Everyday LC**
Largest Divisible Subset
Morning Bi-weekly contest 28
Final Prices With a Special Discount in a Shop (3 points)

Subrectangle Queries (4 points)

Find Two Non-overlapping Sub-arrays Each With Target Sum (5 points)

Allocate Mailboxes (7 points)
**LC 1031. Maximum Sum of Two Non-Overlapping Subarrays**
**LC 1477　(check this LC order : 560 - 523 - 1477) 可以看LC 中国**

1. Find the subarray whose sum is equals to a given target:  check 523. Continuous Subarray Sum
    a. Put (0, -1) at the head of hashMap  is beautiful
    **b. A proof sketch:**

**Suppose `sum_i` represents the running sum starting from index 0 and ending at i, once we find a mod that has been seen, say modk, we have:**
**current one: `sum_i = m*k + modk`**
**previous one: `sum_j = n*k + modk`**
**Thus,**
**`sum_i - sum_j = (m - n) *k`**

2.

## Night Contest (做了2道)

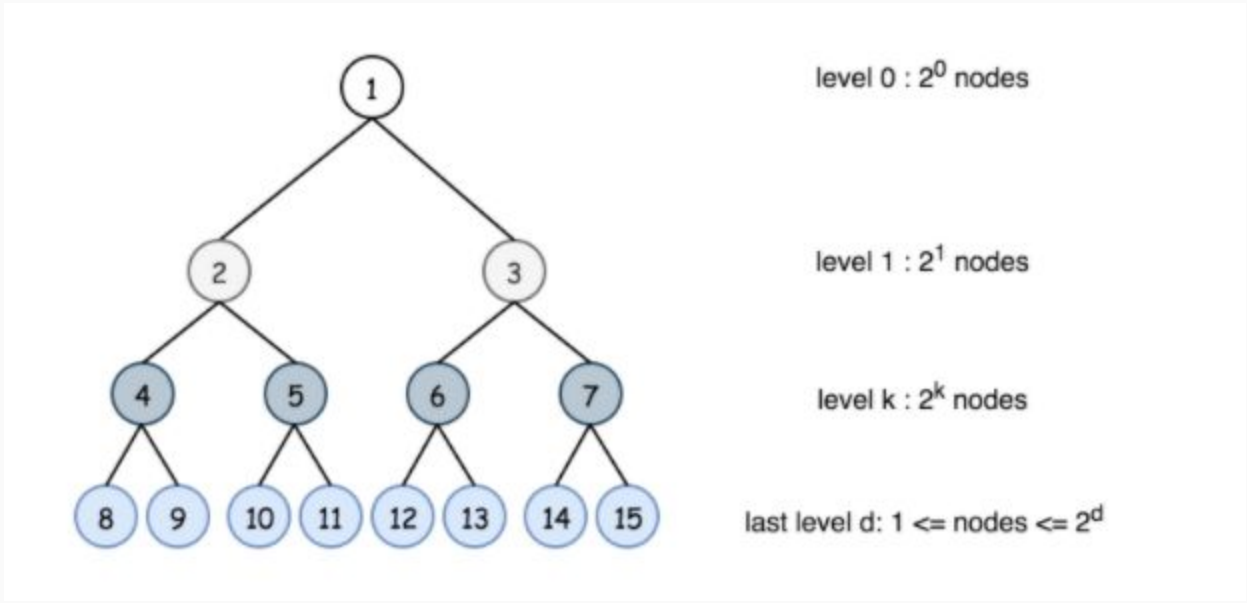
- **Running Sum of 1d Array**[3]
- **Least Number of Unique Integers after K Removals**[4]
- **Minimum Number of Days to Make m Bouquets**[5]
- **Kth Ancestor of a Tree Node**

# June 17  Wednesday
# LC 130. Surrounded Regions  dfs

# June 23  Tuesday

## LC everyday 222. Count Complete Tree Nodes

In a complete binary tree every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible.




## June 24 Wednesday:

LC everyday  96. Unique Binary Search Trees
dp

## June 25 Thursday

## Floyd's Tortoise and Hare

Two pointers ,  slow and fast pointers to find the duplicate or start point
LC 287. Find the Duplicate Number
LC 142. Linked List Cycle II
没太明白

## June 26 Friday

129. Sum Root to Leaf Numbers

**Pair Class in Java**

pair.getKey(), pair.getValue() 查一下inorder的那个怎么写的

Preorder (iterative, recursive, Morris Preorder Traversal. ), inorder, postorder

## June 27 Saturday

**Biweekly contest**

**Topological sort**

Parallel Courses II course scheduler

Everyday Perfect Squares