

1. add Operator modified

- what is the time complexity?

leetcode 282,

Input: num = "123", target = 6

Output: ["1+2+3", "123"]

But you need to consider / for the follow up.

2. 切圆n刀，最多切成多少块

3. DNA_Sequence_in_order

Find the repeated 10-letter DNA sequence, and return a List, and it should be in order, A is before C G T.

What is the time complexity of your method.

4. Evaluate_Reverse_Polish_Notation_modified

Consider factorial !.

5. Find Friends

已知一个函数，输入用户ID，可以返回该用户的所有友好（degree 1 friends），按好友ID从小到大排序。

要求实现函数来输出返回一个用户的所有好友的好友(degree 2 friends), 以及 degree 3 friends。

```
private List<Integer> getFriends(int id){ pass;}
implement
public List<Integer> secondDegree(int id)
public List<Integer> thirdDegree(int id)
```

6. 205. Isomorphic Strings

Question;

第三题：205的follow up，给一个string数组，将iso的词归在一组

{'fff','abc','foo','haa','www','vvv'}-> {{'fff','www','vvv'}, {'haa','foo'}, {'abc'}}

7. 153. Find Minimum in Rotated Sorted Array

Input: [3,4,5,1,2]

Output: 1

Input: [4,5,6,7,0,1,2]

Output: 0

8. FlattenMetaList

Question:

2. 双向链表，但是每一个点还可以有up，down pointer，已知一个链表里没有环，要求把这个链表变成标准双向链表，
每个点的具体位置排列无所谓。楼主开始反应是递归，写好后面面试官说优化一下，，空间要求是constant space，
然后尽管面试官一直在提示tail recursion，还是没想出来（据说地里有原题，可惜当时楼主没看到。。。跪了= =!）

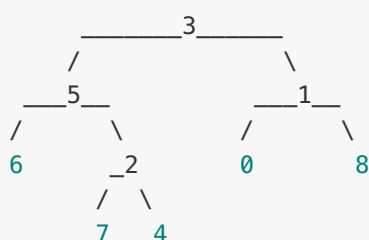
time $O(n)$;

9. Generic_Class_Permutation

Same with permutation, but the type is not int, but a generic T.

10. 380. Insert Delete GetRandom $O(1)$

11. 236. Lowest Common Ancestor of a Binary Tree



follow up, what if node might not exist in this tree.

12. MergeKArrays

```
public List mergeKArrays(List< > input) {}
```

13. MidStack

自己实现一个stack, 可以O(1)实现push, pull, getMiddle, 讨论了几分钟写完又跑了一下test。

14. multidimensional array sum

第二道是新题, 给multidimensional array, 给一个function, 输入这个array以及各个dimension上的index, 可以output这个位置上的数字。

写一个function, input是multidimensional array, 以及array的dimensions, 只能调用上面给的那个function, 输出这个array里面所有的数字的和。

题不难, 是我当时脑子懵了, 一直在想怎么找这个array的各个dimension上的boundary, 其实input就给了。和面试官一直在交流, 但我没说好, 十几分钟一直在纠结这个问题。

后来面试官举了个例子, 立刻反应过来了。但也没有什么时间, 就草草的说了下pseudo code, 用dfs做所有dimension上的不同index的combination, 然后调用那个function求和。

```
interface MultiDimArray {
    int getValue(int[] indexes);
}

public class Multidimensional_Array {
    int getSum(MultiDimArray arr, int[] dim) {
        if (dim.length == 0) return 0;
        return getSum(arr, dim, 0);
    }

    private int getSum(MultiDimArray arr, int[] dim, int pos) {
        int sum = 0;
        if (pos >= dim.length) return 0;

        for (int i = 1; i <= dim[pos]; i++) {
            dim[pos] -= i;
            sum += arr.getValue(dim) + getSum(arr, dim, pos + 1);
            dim[pos] += i;
        }
    }
}
```

```
        return sum;
    }
}
```

15. Path Sum modified

Path Sum的变种：给一个二叉树，每个节点的value都是一位数，比如1,2,3,null,4，那path就是1->2->4和1->3，那return 124+13;

16. Replace_String

17. reverse_string_keep_blanks

18. Shortest_words_threedWords

19. Three_max_min_abs

Question:

Given three sorted arrays A[], B[] and C[], find 3 elements i, j and k from A, B and C respectively such that

$\max(\text{abs}(A[i] - B[j]), \text{abs}(B[j] - C[k]), \text{abs}(C[k] - A[i]))$ is minimized. Here abs() indicates absolute value.

Example :

Input:

A[] = {1, 4, 10}

B[] = {2, 15, 20}

C[] = {10, 12}

Output: 10 15 10. 10 from A, 15 from B and 10 from C

Input: A[] = {20, 24, 100} B[] = {2, 19, 22, 79, 800} C[] = {10, 12, 23, 24, 119} Output: 24 22 23. 24 from A, 22 from B and 23 from C

Solution;

use three pointers

O(n);

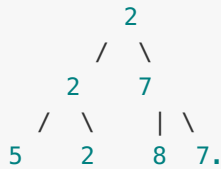
20. Tournament_tree_SecMin

Question:

second min modified

Tournament tree 找secMin;

Tournament tree 的定义是parent 是孩子node的最小值， 如下例 return 5



然后我问小哥要提示，小哥说第二名只能被最后的冠军打败。

所以我就想到只需要考虑被root打败过的node就可以了，就想到了 $O(\log n)$ 的解法，写出来了
思路就是第二小的一定是和最小的有一次比较，所以只遍历有最小值的那条路径，找到路径里面除了这个最小的之外，其他最小的就好了
follow up是找第三小的和找任意第k小的。

21. Question: 2-color

问一个图能否用两种颜色涂色，要求是每条边的两个顶点颜色不同。

22. valid_parenthese_modified

Question:

看括号是否是闭合的

)000 ----> true

(+1^\$#)(#\$) ----> true

)(----->false

()#%33 ----->false

.