11.23.2021

1.leetcode two sum

Hashmap got a set to store the number

Kind of like seeking in the array space

Don’t need the copy of the array to do loop

So the big O time notation is N

1. Leetcode removeDuplicate

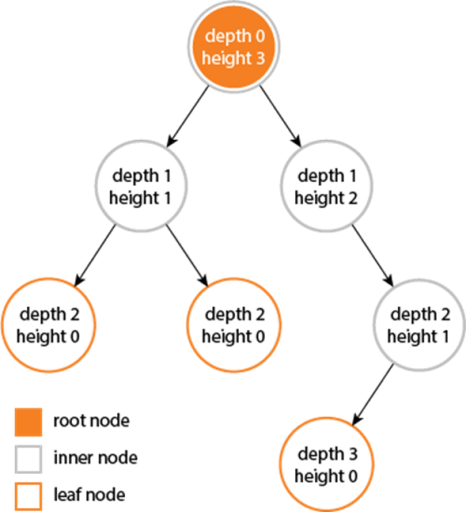
Orderd array use array is the fastest method

to exchange element

double pointer

增强for 循环 自增自减 可以提高时间效率

1. binary search
2. high low when high equals low equals mid either it found or not it’s real
3. 99999 10000 变化的变量定义在for 循环体里面
4. 动态规划方程 滚动数组 同步i 最大和的子数组
5. 合并两个有序升序数组 双指针 取出最小的值 进入队列有序数组转为二叉树 建立 help（）递归 二分



1. 杨辉三角2 滚动数组

第i行第j个element 等于 i-1行的第j和第 j-1的和

Arryllist.add(0) 开辟空间 二次循环从后面写入

1. 时间o 常数时间操作的时间 常以操作次数
2. 空间o 与功能无关 为了完成流程 自己开辟的空间
3. Int找到最右侧的1 取反加一 与上自己
4. Reverse a headlist create node pre and next

While head != 0 loop next = head.next;

Head.next=pre; head= pre; head = next;

Next 肯定指向下一个，head 也指向下一个 ，pre 修改当前的head指针

并且pre指向当前的 head

13 强大之处在于开辟额外的空间 notebook

双链表实现 队列 和 栈 队列 pop push size limit next popint

循环

14 最小值 两个栈

15：leetcode

17：leetcode 找出数组的重复元素 可以使用双指针

18：数组区间可以用low 和 high进行同向遍历

19：循环用来标价的变量 记住 赋值要放循环外面

20： 前三最大元素 if(nums[i]!=nums[i-1]&&++diff==3)

Return nums[i]

考虑目标需要的存储空间是多少

2021.12.7

全正数的数组寻找缺失的数字

可以利用下标和全取负重复利用数组原来的空间

2021.12.8

按照题目要求 夺走几次流程 发现规律

让计算机 执行规律

每次操作既可以理解为使 n-1*n*−1 个元素增加 11，也可以理解使 11 个元素减少 11。显然，后者更利于我们的计算。

12.9.2021

双指针leetcod有序数组 去重可以利用数组抹除原来的元素

结合自己的思维 利用计算机的logic 循环 。。。边界条件

控制语句.

Binarytree inferior nodes have two children and all leaves have the same depth

2021.12.17

Leetcode

单调栈适用于寻找右侧第一个比当前元素大的 值

2021.12.18

输出键盘行 的单词

遍历 输出 nothing new

12.19.2021

数组拆分 min

(a1,b1)贪心也没问题，该问题满足最优子结构，并且每一步的贪心选择可以达到最终整体最优解，每一次的选择只依赖于之前已做的选择，不依赖于后面要做出的选择）。最优子结构你可以把问题规模缩小，比如把排序后的最后两个数去掉，这样2n-2个数的序列的答案，包含在2n个数序列的答案中。 当然，类似于官方解答的证明可以看这个老哥的，写得很好懂

2021.12.27

寻找数组中最大的三个数的乘积

class Solution {

    public int maximumProduct(int[] nums) {

        Arrays.sort(nums);

        int n = nums.length;

        return Math.max(nums[0] \* nums[1] \* nums[n - 1], nums[n - 3] \* nums[n - 2] \* nums[n - 1]);

    }

}

分情况考虑

2022.1.1

Leetcode 滑动窗口 的理念就是下一个i 可以由i-1来获得

当现有循环无法满足条件的时候，开辟额外空间 flag 之类

2022.1.27

#### Leetcode [674. Longest Continuous Increasing Subsequence](https://leetcode-cn.com/problems/longest-continuous-increasing-subsequence/)

难度简单232收藏分享切换为中文接收动态反馈

Given an unsorted array of integers nums, return the length of the longest ****continuous increasing subsequence**** (i.e. subarray). The subsequence must be ****strictly**** increasing.

A ****continuous increasing subsequence**** is defined by two indices l and r (l < r) such that it is [nums[l], nums[l + 1], ..., nums[r - 1], nums[r]] and for each l <= i < r, nums[i] < nums[i + 1].

****Example 1:****

滑动窗口中维护一个最大变量、

2022.4.13

二分方法 小于等于 只有一个元素的时候 要能启动 while(left<=right)