

- leetcode 346, follow up: 要算标准差怎么做 (利用 $\text{Var}[X] = (E[X]^2) - (E[X^2])$)
- leetcode 641, follow up: 多线程怎么做 (只答了用一个mutex把整个数据结构锁起来)
- leetcode 47
- leetcode 315
- 平面上N个点, 找出所有的正方形, 时间复杂度越低越好。
- 给定一串数和sliding window的window size, 要求每次 $O(1)$ 计算mean和std
- 每次读入一个左括号或右括, 判断当前字符串是否合法, 答完用栈的方法之后的变种: 要求 $O(1)$ extra memory, 双向插入括号
- 小车从给定起点到终点运货, 给定一个graph, 每一条边有最大运货的限制, 同时限制小车走过边的条数, 问小车最大的运货量。因为函数接口给的是当前的node, 所以我想的是用二分最大运货量, bfs看能不能在规定步数内走通, 后续考了二分的细节, bfs减少while循环的优化
- leetcode 265
- 给一个binary tree, 如何在没有额外变长空间分配下把它按照in order的方式变成doubly linked list
- leetcode 109, 把双向链表转化为二叉平衡数, 写出了 $O(n \log n)$ 的算法, followup 问怎么优化到 $O(n)$, 应该是先把二叉平衡数存入数组就可以了。
- leetcode 773, 华容道移动, 给出棋盘的初始状态和结束状态, 求移动的最少步数。基本是 leetcode 773题, 只是修改了一下棋盘的大小, 变成了44。一开始用的是BFS的方式写的, 面试官提示了程序中的一点错误, 修改了之后问了下优化的方法, 正好这学期的AI课程学了一些search的方法, 然后就说了下用双向BFS来节省空间, A的方法来进一步减少分支。然后又问了下这道题用A的话使用什么heuristics比较好, 回答了对应点的hamming距离之和。之后面试官说A的方法就不写了, 然后就到第二题了。一个4*4的棋盘, O代表空位, X或者Y代表棋子。每次可以移动一步, 求问最少多少次可以移动至胜利。胜利的条件是, 存在排成一列的或者一行或者一条斜线的X(或者Y)。

OXXY

XOOY

XXXY

YYOO.

我先说了DFS, 后来觉得应该是BFS, 通过Hash值来保存棋盘的状态。

- 一个non-decreasing sorted array 有k个index 上数字变成了错误的 并且 $k \ll N$, 问怎么把他恢复成正确的顺序, 当然不能 $n \log n$ 的重新sort了
- 判断两条线段是否相交, 先求交点, 然后判断交点范围可解, 可以思考的是处理斜率为无限大的直线的时候如何简化代码....
- 给一个数组nums, 它有很多个子数组 (subarray, 要求连续), 每个子数组有一个最小值, 求所有子数组的最小值的和。比如给定 $\text{nums} = [1, 2]$, 所有子数组包括 $[1]$ $[2]$ $[1, 2]$, 则最小值求和是 $1 + 2 + 1 = 4$

- 给m个数组，每个数组里有n个值，都是按照升序排好了。现在从每个数组里分别取1个值，并对这些值求和，这个和记为y。求前x个最小的y。比如：给定数组 num1 = [1,2,3] num2 = [4,9,10] x=2，则最小的两个值是： $1+4 = 5$ $2+4 = 6$
- Implement a fixed size queue，支持pop(), push(), 如果超过capacity就直接return false
- leetcode 56, followup: 给两个Array of intervals，每个array内部是无序的，也可能有overlap。求这两个array所覆盖的区域之间有没有overlap
- 给出Union tree的定义：自己和所有child的value都一样。求一个tree里面有多少个union tree，比如以下这个就有6个： 1 1 2 1 1 2 2 除了顶点的1以外，以其他点为顶点的tree都符合Union tree
- 问了Map和HashMap的实现，并问了HashMap的地址冲突的解决方案。还问了当HashMap中的地址冲突是用开放地址法解决的时候，删除一个key时的操作
- 合并n个BST。只让我写了一个TreeNode to array的函数。
- 给出tree的一个节点的weight的定义: the biggest depth of nodes whose values are greater than A。求一个Tree的每一个node的weight值
- number of island， number of island。面试官给的hint是可以认为向右是*k向左是+1，（感觉和我想的是一个意思？）这样相当于算出一个hash值存储空间可以变少…他给我的hint有minor mistake，他希望我能指出他的错误然而我并没有
- 变种的wildcard matching加follow up
- 給A, B 兩個integer array, 找所有數字的中間值
- leetcode 200
- leetcode 20变种，仅有小括号，先用的stack，后面问如何优化O space
- 有n个城市，每个城市之间可能有路径相连，如果有路径，这个路径会有一个weight，代表的是最大能够承受的车的重量，另外有不同重量的车，能走的路不能超过最大承重，每条路通过路费1元，预算k元，写函数求能够从s开到t的车的最大重量。follow up，如何优化
- 将多叉树转成二叉树，根据他给的例子分析转换规则，并实现，要求不能复制所有节点（就是不能遍历之后重建一棵树）
- leetcode 431
- leetcode 146, O (1) both
- 给一个int[]，求所有subarray中的最小值的和。比如[8, 3, 5, 7], subarray有[8], [3], [5], [7], [8, 3], [3, 5], [5, 7], [8, 3, 5], [3, 5, 7], 最小值分别是8, 3, 5, 7, 3, 3, 5, 3, 3, 返回这几个数字的和40。最优解是O(N)，记录在每个position，左边和右边比它小的元素的个数。应该是立扣原题？希望知道的同学可以告知题号。第二道是利口就灵气
- 给一个BST，有两个node的值交换了，return 正确的BST
- 假设有很多张图片，图片之间有遮挡关系，找到最底层的图片 follow up1: 假设所有的图片都是矩形，给你俯视图，如何找到图片之间的遮挡关系，提一下就好，不用写 follow up2: 给出一个possible的从最底层到最外层的图片顺序，就是topological sorting，因为我当时时间还够就写了一下
- 假设从原点出发，可以往上左右三个方向走，不能经过已经走过的节点，一共走了n步，问总共有多少种路径。给定n步，现在从原点出发，只能往上左右三个方向走，不能经过已经经过的节点。问有多少可能的路径？dp_i表示走n步的可能性。考虑在i-1步的情况，第i-1步是往上，那么接下来就有三种可能性，而且前面i-2步可以以任意方式到达。因此是dp[i-2]*3, i-1步剩下的情况，就是往左往右，对于下一步i都只有两种选择，因此(dp[i-1]-dp[i-2])*2,最后上面的相加就行。

- 一个排序后的长度为N的int list, e.g.[1,2,3,4,5,6,7,8], 改变其中k个数字的位置, e.g. [1,3,2,4,5,7,6,8] (k=2)。要求返回正确顺序的array, 时间复杂度小于 $N\log N$, $k \ll N$, 不要求 inplace。说了个思路: 先traverse, 找到所有位置不正确的数字, 并且从原list中去除, 然后用 Arrays (我用的Java) 里的binarySearch找到这些数字应有的位置。面试官说方向正确, 第二部分时间复杂度是 $k\log N$ 满足条件, 但第一部分最糟糕的时间复杂度是 kN , $k < \log N$ 不一定成立。聊天的时忘记问这个题的思路了==讨论时我问如何确定哪些数字是位置错误的, 比如 [1,3,2,4,5,7,6,8], 是3, 7, 还是2, 6, 面试官说都可以, 但选出的错误数字的个数和k有关。听起来是个很关键的hint, 但是慧根不够, 从这个角度出发我没有任何想法。长度为N的单调递增数组其中有k个元素被修改成其他数($k \ll N$) 要求将整个数组重新排序 时间不超过 $O(N)$. 在小哥的疯狂提示下 总算写出来了 Stack+MergeSort. 忘了说 k是未知的。给你一个数组, 这个数组一开始是排序好的(升序), 这时候替换中间的k个数字, 随机替换, 没有规律。让你在找到比 $n\log n$ 要小的解法, 使得数组重新升序。先说我的做法: 各种纠结之后, 我的做法是, 没发现一个异常点就重新开一个数组, 这样遍历一遍后就会得到至多k个升序数组, 然后就merge k sorted list就好, 复杂度是 $n\log k$ 。面试官提示下口述的做法: 扫描, 发现异常点以后, 就把这个点和之前的那个点拿出来, 这样扫描一遍后就会得到两个数组, 一个是已经升序的, 另外一个为2k未排序的。 $k\log k$ 排序乱序短数组, 然后merge 2 sorted list。给你一个有序数组, 随机替换中间的k个数, 得到一个无序的数组。要求找到比 $n\log n$ 要小的解法, 使得数组重新有序。实际上就是k个有序数组排在一起, 然后merge k sorted lists, 问了时间空间复杂度。给一个长度为n的数字阵列arr, 已知阵列中除了某k个数字被替换过, 其余的数字是已序。example: $n = 10, k = 2, arr = [0, 1, 2, "1", 4, 5, "10", 7, 8, 9]$ ("1"中的数字为被替换过的数字)。并且假设k远小于n, 要求用低于 $O(n\lg n)$ 的做法把阵列排序。解法: 可以当做k个sorted array做merge, 复杂度为 $O(n\lg k)$
- 怎么验证一个变量是空对象: {}, 我舍友说是!!就可以; 第二个问题: 怎么获取数据: 可以用框架可以原生js: 原生js写xmlhttp? angularjs \$http, 还有jquery的ajax; 第三个问题: 有多个url, 怎么快速向这么多个url中获取数据。。。我真的不知道。。。
- merge k binary search tree
- 给定若干点坐标, 连成折线。要求将折线长度k等分, 返回等分点的坐标数组列表。
- leetcode wildcard matching
- 面试官说现在给你一个字符串, 你可以增加或者删除, 要尽可能的balance, 也就是和原来的字符串要尽可能的相似, 你要怎么做(这里我们认为添加是+1 删除是-1 然后想办法让操作完之后接近0), 要返回回文字符串 一开始说dp, 面试官说, 别别别, 直接暴力, 我们暴力来看。然后我给他我说可以用DFS做, 他说可以, 不过我们简单一点, 先考虑删除怎么做。然后我就写了删除怎么做。之后他说可以, 那我们再考虑添加怎么做。我就又写了个添加怎么做。最后就是合并两个, 但是时间不太够了, 就大致说了一下。分析了复杂度(时间和空间)
- 做题只做了一个, 比如abcdabefexy分割成abcdab efe x y, 和其他的子字符串没有重复字母, 思路就是存每个字母最右边的index, 然后遍历数组, 不断扩大右边界, 如果发现合法字符串, 就截断。
- size为k的buffer, 实现这个数据结构。
- 字符串a到b转换, 每次要转换所有字母到另一个, 允许用一个special char做转换桥梁, 问最少转换次数from a to b
- 给一个迷宫, 有门有钥匙, a钥匙开A锁, 求收集齐所有钥匙后从起点到终点的最少步数。给了一个迷宫, 要求求出从给定的起点走到终点的最短路径。迷宫中会有wall, key和locker。key是26个小写字母, locker是26个大写字母, 遇到key要收集, 遇到locker要用收集到的对应key打开, 如果没有收集到对应的key, 就无法通过locker。走到终点的时候必须收集到迷宫里面的所有key。楼主用bfs做的, 讨论了很久如何保存中间状态, 发现用bit来保存key的状态比较好。面试官各种提示,

最后还没写完，所以觉得应该跪了。没想到还是发了offer。

- 一个双向链表和二叉搜索树互相转化，用递归，每次找到那个root节点然后对左右子树继续进行递归就好，不过最好先遍历一遍链表，然后用数组存，每次递归传入这个数组和对应的index范围，最后时间复杂度 $O(N)$ ，空间复杂度也是，反向转化的话也是类似思路
- 染色问题，是leetcode265原题，用dp解决，具体方法可以直接去leetcode上看。写之前说了下思路，问了下空间复杂度。
- 经典n-puzzle问题，但是面试的题目变成nm的puzzle，follow-up是如何判断解是否存在。解法: A* algorithm, follow up可以在 $O(nm)$ 的时间根据奇偶性判断(<https://www.cs.bham.ac.uk/~mdr/t...lesSolvability.html>)。我没有解出follow-up，但是面试官说没关系只是要看思考逻辑。follow-up复杂度写错了，应该是 $O((nm)^2)$ 或是 $O(nm \lg(nm))$ 才對
- 给定一个矩阵，找到两个元素，要求一个在左上方，一个在右下方，使得两个元素差值最大。
- 给了一个字符串，包含了若干大写字母和小写字母，然后要求将字符串的小写字母移到字符串的前面，大写字母移到字符串的后面，顺序无所谓。用了两个指针来做，第一个指针遍历整个字符串，第二个指针记录index最小的大写字母，每次遍历到小写字母的时候，如果index大于指针记录的大写字母的位置就交换，然后再去找下一个大写字母。
- 给了一列数字，要求返回每个数字的左边的最后一个比他大的数字的坐标，如果没有就返回-1。例如5 4 2 1 3，就返回-1 0 1 2 1。有点类似leetcode的739题，主要是用stack来解决。一开始是用顺序遍历来做的，然后面试官给了一个反例，想了一会就在面试官要提示的时候想出来应该逆序来操作，然后改了下代码写出来了。想了一下，stack那道题顺序遍历也是可以的，当遍历的数字比stack顶端小的时候就压进stack，大于等于栈顶的时候就pop掉顶部元素。可能当时有点紧张，思路不太对。
- kdtree
- 一个游戏装备，每一轮附魔升级有一定的成功和失败几率，需要升到最大级别的期望次数，我给了个dp的解，他说跟想让我用markov解，但是这个也行。
- 给一堆点，已知他们是从一个正态分布生成的，估计平均；
- 有向图找圈，必须要用topological sort
- 给一个pixel matrix，matrix可以理解为照相得到，不同颜色的pixel组成不同的广告牌，判断广告牌之间的遮挡关系 如下

[1,1,1,1]

[1,1,2,2]

[1,1,2,3]

不同的数字代表不同的颜色。例如此表示有

[1,1,1,1]

[1,1,1,1]

[1,1,1,1]

[2,2]

[2,2]

[3]

3种广告牌，而1被2所遮挡，2被3所遮挡，所以在照片中的pixel分布为：

[1,1,1,1]

[1,1,2,2]

[1,1,2,3]

求距离最远的广告牌的颜色。

给一个图像，上面有若干矩形，矩形颜色各不相同，可能有重叠。可以想象成一摞带颜色的矩形无规律放在桌面上，然后从上往下看。问可能有哪些颜色的矩形可能是在最底下一层的。注意是可能。比如：BBP BWG RGG 这时候可能的颜色是，B P R G

- multiple inheritance diamond problem
- leetcode 907，要求解到单调栈的解法， $O(n)$ 复杂度。可与leetcode 84题一起练习。
- 然后问了一个栈是不是线程安全的。。我拍脑子就说不是，结果一查继承vector是安全的，面试官也没指出来。。
- str1 变到 str2，求最少的变次数。规则1：a -> x的话，所有的a都需要变成x（x表示任意其他字母）规则2：a->, ->x
"abc" -> "bbc" // 1, a->b "aba" -> "bbc" // -1, not possible "aba" -> "bab" // 3, a->*, b->a, *->b
- 给文件列表List files, 实现一个gitignore, 可以过滤掉gitignore里出现的文件，需要满足gitignore的规则。gitignore规则：如果是*, 则表示通配0个或者任意字符。比如 /path/to/*file. /path/to/abcdefile满足(任意), /path/to/file满足 (0)。如果是?, 则表示通配一个字符。比如/path/to/?file. /path/to/zfile, /path/to/xfile等满足。如果是[], 则表示满足括号里的任意一个字符。比如/path/to/[abcde]file, 则/path/to/afile, /path/to/bfile等满足。重点是, "/"不能匹配难炸了。跟leetcode 10有点像，但是比这个难多了。写了半个小时放弃。写了写可能的思路(dp)。后来面试官告诉我用正则表达式API。
- 实现deque
- 给定n个圆柱体，半径和高，选K个使累加表面积最大
- C++11,14,17的特性你知道有哪些吗？我：emmmm。这个不会，没关系，你知道智能指针吗？
- 题目是给一个N-ary tree，如何将其转化为一个二叉树？转化规则为左孩子，右兄弟，就是左儿子不变，右孩子变为自己右侧最近的那个非空兄弟。follow up:任意给一个二叉树，可以还原成N-ary tree吗？如果不能，满足什么样条件的可以呢？？原题我是用hashmap+level-order traversal做的，然后follow up是不能，需要满足根节点没有右子树且其任意节点的右儿子深度不超过N，第二个条件我开始也没想出来，我可能表达的也不是很准确，其实就是root = root->right，然后直到null停下来，这个深度不能超过N，因为子节点数量有限。
- 给一个正整数集合，求一个和最大且能被3整除的子集。Follow up: 如果集合里有正有负，怎么做。用DFS是肯定可以做的，但我当时想的是先排个序，然后greedy地取集合里的所有数，看看除3余几 1) 如果余0直接return 2) 如果余1，考虑是丢掉一个最小的除3余1的数，还是丢掉两个最小的除3余2的数 3) 如果余2也是类似的 后来跟面试官讨论发现其实不用排序。。打个擂台就能找了（但其实我是想着排序了代码好写一点orz）优化后时间复杂度是 $O(n)$ ，本来还担心这个方法会不会有点野鸡，但是讲道理效率确实比DFS好得多。。。 follow up: 加入负数的话也是类似的，一开始greedy地取所有正数，然后再考虑是丢掉最小的正数还是加入最大的负数，复杂度一样
- leetcode 136变体，如果出现两次的数总是相邻的，有没有比异或一遍更高效的做法。要把 $O(n)$ 的复杂度继续优化，显然用Binary search，结合位置的奇偶性就可以了。
- sliding window minimum，给一个指定size的sliding window，返回每一次的最小值 面完看了一下网上有maximum的题，也不是很难，可以做到 $O(n)$ 时间

- 给定两个string str1, str2, 判断能否将str1转换成str2, 每个字符只能转换一次, 每次转换所有相同字符, 如 "abc" -> "def" return true, "a" -> "d", "b" -> "e", "c" -> "f" "abb" -> "bcc" return true, "b" -> "c", "a" -> "b" "abb" -> "baa" false, 无法转换-baidu 1point3acres
follow up 1: 给定一个符号"", 字符可以先转换成""再转换成其他字符, "abb" -> "baa" return true, "a" -> "", "b" -> "a", "" -> "b".
follow up 2: 没有"*, 不限字符转换次数, 即可以先将一个字符转换成中间字符, 再对中间字符转换。"abb" -> "baa" return true, "a" -> "c", "b" -> "a", "c" -> "b"
- sorted linked list to BST
- LRU Leetcode 146.
- sliding window, 实现insert, mean, std方法。要求指定window size, 插入时若window满, 删去最早insert元素。mean返回均值, std返回标准差。Follow up: O(1) 实现 mean(), std()
- 面了1题, 图论。已挂。你在开一辆车, 给定车的重量。给出一个起点和一个终点, 不同的路用不同的最大承重量。你用n个硬币, 每通过一条路就要支付一个硬币。求能从起点到终点的最短路径(每条路长度都是1)。
- 双向链表里面存的非减序列 要求转为平衡二叉搜索树。要求: 自己写数据结构(树节点结构类似于双向链表 树左右 类似于node pre post) in-place 不能创建额外空间 可以用hashmap。follow up: 反过来 树转链表 要求相同。
- leetcode 4
- 给一个最小堆, 每一个最小堆的节点都唯一对应一个图节点, 让设计一个数据结构, 使得可以修改最小堆节点的值而不改变对应关系
- leetcode 99 我先提出可以中序遍历存下BST里的所有数, 改完之后再生成一个新的, 然后说这个复杂度高, 提出更优的做法。问了复杂度
- 给一个电话数字键盘, 1-9的九宫格, 然后拨号方式是马走日(下一个数字相对上一个数字的位置是个“日”字), 问你拨n位共有多少种可能。刚开始想的还是BFS(对不起我爱BFS爱的深沉, 感觉没啥搜索的题目是不能用BFS解的), 可以拓展出所有可能。后来改进了改成DP, DP方程还是很好想的, 就不贴了(f[i][j]表示已拨了i个数字且最后一个数字是j的可能性共有多少), 不过有趣的是面试官似乎和我想的并不是同一个方程, 中间我写了一个bug, 然后他觉得不应该改, 然后我们就应不应该改讨论了一会。
- 有一辆小车沿x轴运动, 有两种情况会改变其速度, 第一种是到达某个时间点, 第二种是到达某个地点。求每个时间点所对应的x轴位移。递归解决。
- sqrt double (binary search)
- calculate possibilities (dfs -> recursion -> dp)给你n个硬币, 每个硬币正面向上概率为一个array [p1, p2, ..., pn], 问题: 每个硬币扔一次, 计算k个正面向上的概率
- 给一棵二叉树, 然后每个节点多个属性叫sibling, 指向同层右边第一个节点, 如果右边没节点就是null。给出二叉树, 让补全每个节点的sibling属性。思路是用BFS(其实就是层次遍历), 然后每个节点去连队列里同一层的下一个节点。Follow up是O(1)空间, 看下leetcode题解好像是有有一个指针在上一层, 有一个指针在下一层, 然后一边遍历一边把下层的节点的sibling直接连起来, 这样虽然不维护一个实际的队列, 但是在访问某个节点的时候, 下一个节点访问的是他的sibling, 其实和层次遍历的拓展顺序是一样的。
- 给定一个BST。implement BST节点的get_weight方法, 其中weight定义为此二叉树中value大于此节点value 的其他所有节点的最大深度

input:

5

/\

3 7

/\

1 4

返回

2

/\

3 -1(因为7为最大)

/\

3 2

given a BST, weight(node i) = max depth of node j, where i.val < j.val

val: 3 /\ 2 5. 1point3acres // 1 4 weight: 3 /\ 3 -1 // 3 2

- Description

Given two alphabet strings str1 and str2. You can change the characters in str1 to any alphabet characters in order to transform str1 to str2. One restriction is, in each operation, you should change all the same characters simultaneously.

What's more, you may use another special character * if necessary, which means during each operations, you may change the alphabet characters to *, or change each * to a specific character.

We want to know the minimum operation times. If impossible, return -1.

Template

```
int MinOpTimes(string str1, string str2) {  
    //...  
};
```

Examples

Example 1:

str1: accs, str2: eeec

operation 1: change 'a'-'>'e'; // str1: eccs

operation 2: change 'c'-'>'e'; // str1: eees

operation 3: change 's'-'>'c'; // str1: eeec

return 3;

Example 2:

str1: accs, str2: efec

return -1;

Example 3:

str1: abb , str2: baa

operation 1: change 'a'-'>'*'; // str1: *bb

operation 2: change 'b'-'>'a'; // str1: *aa

operation 3: change '*'-'>'b'; // str1: baa

return 3;

- web MVC TCP/IP AJAX Socket DOM CSS React 细节, 虚拟 DOM, DOM diff 算法等等
- Continuous Integration 抓取网页信息的js代码, 还有同步异步问题, es6和es7 的解决方案
- AJAX fetch, async await Google s2 算法
- Css JS js 的promise, 异步 实现 lodash 里的 throttle 和 debounce 两个 API 闭包 + setTimeout
- $\text{pow}(x,n)$, $\log(n)$
- longest common subsequence +follow up dp +返回一个结果 +返回所有结果
- 两堆石子(m,n), 两个人A和B, 每次只能取(0,k)或者(k,0)或者(k,k), 其中 $k \leq \min(m,n)$ 。求问如果A先取, A有没有必胜策略。首先想的是迭代, 边界条件是如果 $m=n$, 那么A必定赢了。初步的想法是, $f(m,n) = f(m-k,n-k) \mid \mid f(m-k,n) \mid \mid f(m,n-k)$ 等等 $0 < k < \min(m,n)$ 。时间复杂度是3的 $\min(m,n)$ 次方。后来又问能不能提升复杂度, 想到了记忆化搜索。但是最后面试官说可以达到 $O(m)$ 的复杂度。最后一题忘记说赢的条件了, 就是谁先把两堆石子都取完了, 谁就赢了
- leetcode 72 Edit Distance
- 物体从 $x = 0$ 处开始运动, 初速度 $v_0 = 1\text{m/s}$, 给出两个array
 $l_1 = [[2,4],[4,7]...]$
 $l_2 = [[1,3],[5,8]...]$
 l_1 表示物体运动到 $l_1[i][0]$ 时刻时, 速度变为 $l_1[i][1]$,
 l_2 表示物体运动到 $l_2[i][0]$ 坐标时, 速度变为 $l_2[i][1]$.
给定一个时间点t, 求物体在t时刻的position.
- 给一个size 为n的堆, $O(k \log k)$ 时间求这个堆中前k小的数 How to get the k smallest element in $O(k \log k)$ time
- leetcode 215, 但是不能用priority queue, 必须用quick sort来写
- leetcode 53, followup是找出两个subarray, 使他们sum最大, 这里我用的是两个数组保存每个位置左边的maximum subarray和右边的maximum subarray。然后找两个数组对应位置sum最大的就行了
- 给array of integers里面有一个数字是单独出现 其他都会出现两次(而且一起出现)ex: [1,2,2,3,3]要判断哪个数字是单独出现的,以这个例子的话就是 1, 用XOR 去算吧 剩下的那个就是单独出现的了 复杂度是 $O(N)$.希望再想其他方式可以优化的 比如说 $O(\log N)$ 复杂度.
- 给你一棵树 问是不是uniform tree (也就是 整棵树的值都一样) follow-up: 问这棵树有几个subtree 是uniform
- 给一个数组 里面只有 0和1,问最少次数把0换成1 或把1换成0 可以让 0都在左边 1都在右边 (或者0都在右边 1都在左边). [0,1,0,1]的话就是把第一个1换成0 可以达到分边
- leetcode 76

- 字典树
- 字符串匹配1：正常版 => KMP/RK
- 字符串匹配2：模版串可以shuffle (abc 可以匹配bca)
- 字符串匹配3：模版串可以shuffle + mapping (abc 可以匹配321 abc =》 cba =》 321)
- 字符串匹配4：AC自动机
- 给n个点和m个区间，一个点最多匹配几个区间 问最多匹配几个区间和点：堆+扫描线 需要证明正确性 貌似用二分图？
- 给n的点形成一条折线，求k等分点：二分
- 简历 + 多线程知识 + 实现多线程队列 + OS (进程间通信：管道 命名管道 消息队列 共享内存) 问给offer去不去
- 有n个object 两两之间可以有三种关系(相同种类1,不同种类2,不知道2) 给m个三元组 (object1, object2, relation 1/2/3) 假设不存在矛盾 求问给定一些二元组 (object1, object2) 输出他们的关系
A B 1
A C 1
A D 2
C F 1
D E 2
则输出 B D =》 2 A E =》 3 A F =》 1 并查集建点 (same relation) + 图遍历建边 (different relation)
- n个点在2D平面上，return k等分点的位置并返回。基本就是presum，然后考虑各种corner case, You are given a path consisting of integer points, and you want to cut it into equal k segments. Find endpoints of each segment.第二题写码，给N个点，连起来，长度K等分，找到等分点（在折线上），也不难，就是码有点难写，cornner case, dummy node 比较繁琐。medium 难度，做法就是presum 然后binary search (或者map到index也可以，我用了upperbound) 到位置找点，我没仔细写，写了几行core code，也是O(N)。
- 有n枚硬币，每个硬币掷一次正面朝上的概率各不相同，假定第i个概率为pi。如果把所有硬币都掷一次，求k个硬币正面朝上的概率。一开始想的是DFS，面试小哥说复杂度太高了，后来想到用DP。然后被问复杂度，follow up是问空间复杂度怎么优化到O(n)。
- Find a polynomial time algorithm to determine whether two trees are isomorphic
- 循环队列
- hash map index撞一起怎么办
- 口头design 一个类似于找一对朋友的东西，要求朋友间距离小于K，并且朋友属于同一组，不难，用一个sliding window size K，running time O(N)。
- leetcode 200 follow up 694 253
- docker image和container 问VPN的原理和怎么搭,多线程的概念，问程序运行时calling track怎么打log高效，他的回答是开辟一块同时进行读写操作的内存，先往内存里写然后同时往硬盘读, 为了避免冲突问要怎么加锁，如果整个空间加锁等待时间太长怎么办，然后他跟我讲要对内存进行原子操作？

- One is finding the next greater element
- 一个数轴上，有 n 个点 (x_1, x_2, \dots, x_n) 和 m 个区间 $(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$ 。每一个点只能匹配一个区间，每一个区间也只能匹配一个点。匹配的必要条件是点包含在区间之内。也就是对于 (a_i, b_i) ，当 $a_i \leq x \leq b_i$ 时，可以进行匹配，当然也可以选择不匹配，把这个区间让给其他的点。求最多可以有多少个区间被匹配到。可以抽象成二分图最大匹配，我觉得方法是可行的。如果面试官给我一丁点关于往图上面思考的提示就好了，我应该就能想到。可惜他没有。
- 给定 N ，请通过从小到大输出，分母不超过 N 的真分数序列。比如给定5，就输出 $1/5, 1/4, 1/3, 2/5, \dots$ 等

先开始想了一个用PriorityQueue的算法，因为不超过 N 的序列中，以分母归类，以 $n=5$ 为例，

$1/5 \ 2/5 \ 3/5 \ 4/5$

$1/4 \ 2/4 \ 3/4$

$1/3 \ 2/3$

$1/2$

首先把每一行的第一个元素加入PriorityQueue，再poll出来，poll到的数用一个指针往后移动，这样时间是 $O(n \log n)$ 。后来面试官给提示说，这其实是一个从左到右，从上到下都递增的序列，每次只需要比较右边的和下边的谁大就可以了。