

CHECKLIST

	Date	題目	思路
☑		给一个char矩阵。X不能走O可以走，求从左边到右边最少几步，以及返回路径	求最短路径可以借助queue用BFS实现，每一次用map记录一下当前的点从哪个点过来的，这样就能输出路径了
☑		给一个set问能不能找到四个点组成平行于坐标轴的矩形 followup是如果要求的矩形不一定是平行于坐标轴的？	对于第一个问题直接任意选取两个横纵坐标都不相同的点，然后寻找另外两个点，时间是n^2；对于followp则是任意找三个点并满足勾股定理，然后通过横纵坐标的线性关系找到第四个点,时间是n^3
☐		加入通配符(可以代表任意长度的任何字符串)的LC211add and search word,并且要返回所有满足输入条件的string	遇到统配符就遍历所有非空的子trie将同样的字符串或者去掉当前通配符的字符串递归到下一级处理，退出的条件要么是组装的sb的长度已经超过了dic中所有string的最大长度，要么是扫描通配式的指针已经超出了范围。注意如果遇到结束符并且输入的通配式已经扫描到了末尾说明找到了匹配。
☐		给定一个字符串和一个切分长度k，将一个sentene进行切分，代表最终切分次数的string也标记在每一个切分section的末尾也占长度，求最小切分次数	对最小切分次数做binary search
☑		给定两个压缩表示的vector 比如[(3,2),(4,1),(2,3)] 就是[2, 2, 2, 1, 1, 1, 1, 3, 3]的缩写 现在给你两个这样压缩好的vetcor 让你求内积 当两个长度不同的时候 按最小一个的长度算	每一个vector都用两个指针来表示现在scan到第几个pair的第几个，然后依次相乘加到结果里面
☑		300.Longest Increasing subsequence 的followup, 输出最长的所有元素	
☑		给一个array 比如 [9,9,5,3,3,7,1] 里面的数字都是0到9之间 inclusive 给你个k 给出top k largest number sum 比如 top 3 = 9+9+7	nums中元素范围有限，典型的bucket sort
☑		字符串含+，*和数字，return int结果	直接对加号split然后在每个section中对乘号split。但是要注意两个符号在java中都是特殊符号，所以要首先惊醒替换再计算str.replaceAll("[+]", "+")
☑		给一个代表BTree inOrder Traverse 的array，要求返回一个minheap tree	其实就是LC654的换种说法，每次取当前范围最小的元素作为root，左右子集分别作为左右子树，递归做下去
☐		BST to MIN-Heap	如果允许使用额外空间就先inorder traverse放进list里面然后level by level的将这些node建成一个heap；如果要求in place？
☐		BST to DLL	就是inorder过程中，不断找到中节点左子树最右和右子树最左，连起来；最后反回链表最左即可或者用morris traverse实现严格O(1)空间的转换
☐		BST to circle DLL	唯一不同的就是要用给一个global记录一下tail，最后将tail指一下
☑		Max Sum of Subarray with size k	如果其中全部是正数，可以用双指针，实现时空均最优；如果不只有正数，那么就要用map辅助来实现时间最优
☐		read4k and print line	
☐		Delete Double Linked List Node. Input is head & target(node), delete target, return head.	这道题关键要考虑dll有没有loop。由于dll的特殊性，如果它有环的话只可能是首尾有环。为了能够cover这种情况就不要使用dummy head删除了。分情况讨论：target前面为空，后面为空，后面指向自己(只有一个元素的环)，和普通的情况
☐		给一堆node，判断这一堆node是不是属于一个完整的binary tree	遍历每一个节点，找到这个节点的left和right，并且孩子做为key，当前节点作为value放入到map中，在加入之前只要map里面已经存在这个key了，那么说明有loop。如果能顺利遍历所有的点表明并不存在loop，那么map的size就是边的数目，最后检测一下如果map的size + 1等于输入的node数量那么就说明是完整的binary tree。总结：检验无向图是不是tree的时候可以用过边和点的数量关系加上没有loop或者所有点连通两个条件之一来判断
☐		sorted DDL to BST	
☐		同时判断递增或者递减	用前两个数字确定到底可能是递增还是递减，然后用这个标准去判断结果。注意如果出现两个相邻的值相等，一定false，如果不判断这个会有bug
☑		找离原点最近的k个点，要nlogk的算法，用优先队列，input是(List<Points>, k), output(List<Points>), 有个Point class，求离原点最近的K个点	直接用priorityqueue，为了代码的简洁用一个另外的distance计算距离，trick是queue的长度可以设置为k，并且按照远点距离递减的顺序放入，这样在加入新点且queue满的时候就可以随时保持queue里面的是当前距离最近的k个点。将二维转化为一维之后可以用一维的部分方法：用quickselect可以实现平均O(n)，最差O(n^2)的时间复杂度。另外如果序列已经是sort好的，可以使用binary search实现O(log(N-K))的时间负责度，或者简单的方法用两边双指针实现线性复杂度
☐		给一个字符串比如 12345=67 可以在等号两边随便加加减号使得等式成立，求所有的方法(只有加减号，但是可以在第一位加减号，要返回等式的样子)	分别找左右字符串所有的可能结果，结果的值作为key，对应的所有string表达式的list作为value，存在map里面然后组合两边的结果。注意注意注意敲黑板！02不是合法的切分！！要判断
☑		一个array，有m段递增序列，输出排序好的	转换成merge k sorted的问题，trick就是将所有的切分pivot存入一个set里面，一旦某个组的指针在set里面说明这个组已经遍历完毕了
☑		打印root到leaf的所有路径	类似257,只是将原来的string输出路径改为list输出，退出每层递归之后做一下remove list最后一个元素实现一下backtracking就行了
☑		压缩数字输入1111334222返回41231432	注意边界，判断一下先继点是否存在，跳出循环之后也要append一下
☐		判断两个图是否同构	super难，还没做
☑		200和463的综合：给定条件与前者相同，但不找个数了，要找出周长最大的岛的周长	周长就是岛中1的个数*4 - 右下两个方向邻居1的个数*2.因为1岛的形状不定，所以只能用dfs和global的变量来计算周长，找到每个岛屿的周长并比较
☑		两个BST从小到大输出成一个list，讨论空间复杂度，最好最坏情况	参考LC173的BST Iterator，用两个stack来维护，每次peek两个栈顶元素，选择小的哪一个pop出来放入结果并将其右孩子的所有左节点都放入stack。最终将两个stack中非空的那一个剩下的所有节点按照inorder的顺序放入res
☐		given 2 list of interval representing users online offline timestamp e.g (already sorted), find all intervals that both of users are online.	其实就是找两组interval的所有intersection区间，用两个指针扫描，如果intersect，将交集存入res，剩下的后半部分更新原来end
☐		給2個function: getFriends(a)會回傳a的所有好友，getMutualFriends(a,b)會回傳a和b的共同好友。用這2個function寫一個推薦好友的算法 這題比較像是開放式的問題，先和他討論要如何推薦好友，他覺得OK後我再寫出來， follow-up是sort最後的結果使得好的candidates能排在前面	假设我们为a推荐好友。先求出所有的二度好友：好友的好友去除所有的好友，然后将这些二度好友按照共同好友的数目来排序

	Date	题目	思路
<input type="checkbox"/>		Remove Invalid Parentheses只返回一个结果	stack1存放多余左括号的index, stack2存放多余右括号的index, 如果来了一个右括号但是stack1非空这时直接从stack1里面pop出来一个就好了。最后将两个栈中对应index的括号删除, 注意删除的时候要从前向后, 这样删除之后不会印象后续字符的index。如果要进一步优化空间的话。可以two pass扫描, 第一遍删去所有invalid右括号并记下所有左括号的位置以及一共多出几个左括号, 然后第二次从后向前扫描, 删掉相应个数的左括号
<input type="checkbox"/>		给一个树 (不是二叉树), 找最深叶节点的最近公共祖先。	用BFS从上到下遍历, 遍历的同时用map记录下每一个节点的父亲信息, 由于树的性质, 每个节点只会最多有一个父节点, 所以子节点作为key, 父节点作为valud。然后将最深一层的所有点都放进queue里面。每次都把所有元素pop出来并将他们的父亲push进去, 最终当queue里面只剩下一个元素的时候就是他们的公共祖先啦! 优化方法: 用double linkedlist实现queue的功能, 得到所有的最深节点之后, 只取最左和最右的。然后得到二者的公共祖先
<input checked="" type="checkbox"/>		input string, 去除string中重复的character, 返回一个新的string; abcad, 返回abcd. 如果不确定只有小写字母就用set存就好了	用个set记录一下, 之前没有出现过再append, 或者直接用长度为26的数组记录。虽然空间比map稍稍大, 但是不用用多余的数据结构
<input type="checkbox"/>		返回subtree的最大值	用一个global的max更新
<input type="checkbox"/>		有个interval stream, 顺序是乱的, 写个方法, 每来一个interval, 返回目前为止所有interval合并以后总的长度。讨论了几种做法, 写了insert interval的办法 (存在arraylist里), 聊了用BST的做法, 当时觉得要自己写个BST没法现在写。后来结束以后查了下文档, 发现其实Java也有办法treeset中返回某个元素的iterator的方法 (headSet或者tailSet), 虽然比C++麻烦。	
<input checked="" type="checkbox"/>		输入两个array {3,2,4,1,5,0} {D,C,E, B,F,A} 程序结束后得到 {0,1,2,3,4,5} {A,B,C,D,E,F}	每次交换一对儿, 保证其中一个放在正确的位置上
<input type="checkbox"/>		排序string, 输入List, 输出排序的List e.g. a4b3 (a 是产品名称, 4 是产品版本), a3b2, b1 => a3b2, a4b3, b1; followup, 没有产品名称只有版本, 如 “1”, 答这个会被放在返回List最前面	就直接compareTo? ?
<input type="checkbox"/>		一堆多米诺骨牌 上半边和下半边分别由数字 让你找是否存在一对儿(两片)使得上半边数字的和 和下半边数字的和都等于6	上边值作为key, 所有相同上边值的下边值放进作为value的set中, 之所以用set是基于假设不会出现两个上下片都相同的骨牌, 所以在相同set下的值不会重复, 要注意3+3的特殊情况, 这种情况直接在key为3的set里面找target, 依旧默认上下片不会出现同时为3
<input type="checkbox"/>	11.26	给一些sets, 保证每个set内元素不重复, 如果两个sets有任何共同元素, 则让其合并, 直到不能合并为止。输出最后的sets的状态。 栗子, Input: {1,2},{2,3},{3,4}{5,6,7},{7,8},{9},{9} Output: {1,2,3,4},{5,6,7,8},{9}	用无向连通图来做然后dfs扫一遍的同时加入res。类似323
<input type="checkbox"/>	11.28	task schedule(621)保留task顺序	用一个数组记录一下上一个相同task的位置,用queue来保证顺序
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input checked="" type="checkbox"/>		sorted的array中数target数目	用LC34的方法得到第一个targer的位置和最后一个target位置, 时间O(logN)
<input checked="" type="checkbox"/>		给定一个数组和一个目标值, 数组中均为正数, 要求找出数组中是否含有一个连续子数组的和, 等于目标值, 返回true or false ;follow up 数组中包含负数该如何修改	全部是正数的时候可以用two pointer做到时空最优, 因为sum是递增的; 但是如果 有负数的话就只能用hashmap
<input checked="" type="checkbox"/>	11.26	给一个有正有负的递增数列, 返回一个按绝对值大小排列的数列	从两侧到中间当成两个sorted list做插入即可, 如果要求inplace用quicksort
<input type="checkbox"/>			
<input checked="" type="checkbox"/>	11.26	给一个质数数组 (无重复), 输出数组元素所有可能的乘积	由于数组是质数, 所以这个问题等同于求subset
<input checked="" type="checkbox"/>	11.26	sparse vector dot product	用两个map, key是下标, value是数值。两个map的key相同的地方的数值相乘 加到结果里面
<input checked="" type="checkbox"/>	11.26	奇葩字符串比较, 分数字和字母的block来比, 数字按照数值绝对值大小, 字母按照字母顺序, 数字的block无条件大于字母的。还有很多边界条件需要确定 (比如大小写, 数字block会不会overflow之类	compara to
<input type="checkbox"/>		机器人走房间, move函数就是move这个动作发生了, move(0)如果返回true, 那么机器人已经在上面这个位置了, 不是告诉你能不能走, 而是你已经走了。只有返回false才是在原地, 告诉你这一步走不动。	从假设的(0, 0)点出发, 用dfs, 如果可以走并且没有在set里面那就走, 如果已经在set里面就不要走了退回来。最后set的size就是结果。为了便于返回, 定义一个moveBack的function
<input checked="" type="checkbox"/>		给一个数组, 把它变成二叉树, 该二叉树中序遍历结果和数组一样, 同时还要保证这是一个minstack	每一次选取当前最小元素作为当前subtree的root, 左边所有元素和右边所有元素分别是左子树和右子树, 然后递归的分别找左右子树的最小值, 即root。和LC654一样的思路, 换个说法
<input type="checkbox"/>		链表找环	用快慢指针的方法是1.5N的时间, 如果要实现严格的N时间就要改变输入的结构, 每当扫描一个点就将其next指向head, 如果后面 有任何一个node指向了head, 那么说明有环, 因为前面的所有节点都指向了head, 后面的node指向前面的任何一个节点都会指向head
<input type="checkbox"/>	?	A树中有一个path可以到某个节点a, B树中根据相同的路径找到b. 给出A,B,a 求 b	可以用两个queue, 对A和B同时进行DFS, A找到a的时候B就会找到b.注意BFS好像是不行的.....
<input type="checkbox"/>	11.28	LC621的变种, 给出任务单和同种任务之间的cool down间隔, 要求计算每个任务的执行时间列表。比如任务单为[1, 1, 2, 1], cool down间隔为2, 那么每个任务的执行时间应该是[0, 3, 4, 6]。这题用hashtable可以得出O(n)的解法	就是用array记录一下前一个相同的char的位置, 如果和当前扫描位置间隔大于n+1, 那么放在当前位置就行, 否则要放在上一次位置+n+1的位置
<input type="checkbox"/>	11.29	给出一系列区间, 比如{ [1,3], [3,5], [2,4], [4,7], [4,9], [7,12] }。问如何用最少数目的区间来覆盖目标区间 (比如[2, 9])。在这个例子中, 答案应该是[[2,4], [4,9]]	这道题还要考虑无解的情况.用greedy算法, 先用start对区间进行排序, 找到和target有交集并且能覆盖target的start的所有interval中end最大的, 然后用这个end更新target的start, 然后对更新后的start做同样的操作
<input checked="" type="checkbox"/>	11.26	Sort an input string using an arbitrary alphabet-google For example: custom alphabet: xyzabc input to sort according to custom alphabet: cyxz output according to custom alphabet: xyzc	将word的每一个字符频次用长度为26的bucket记录, 然后遍历alph的每一个字符一次append到sb; 或者将每一个放进map里面, 更节省空间但是多了一个数据结构

