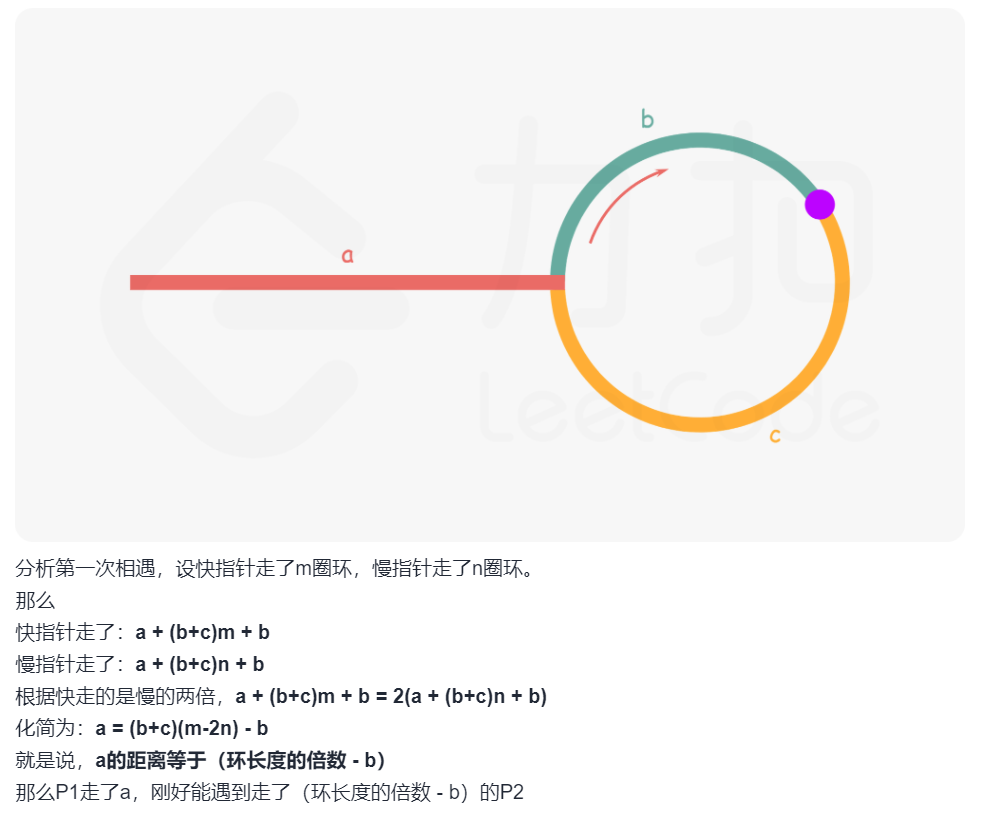
# 面试题0208



N实际上为1



# 面试题0403

广度优先遍历中的层次遍历

# 面试题0205

反向存放好写。

正向存放可以用两个栈存储数字、或将链表反转（头插法即可）。

# 面试题0204

法一：两个链表，一个存储小于x的值，一个存储大于等于x的值。

法二：从头到尾遍历，遇到小于x的值，插入到头部（为了方便，需要建立头节点）。

# 面试题1625

用HashMap，键为key，值为（key，value）组成的节点，HashMap可以保证O(1)的查找时间。同时将（key，value）节点组成双向链表，表示数据在缓存中的顺序，表尾节点是最近被使用过的，双向链表保证了在调整节点顺序时的O(1)时间。

# 面试题0305

用另一个栈辅助操作，so easy

# 面试题0303

用一个列表，列表中每个元素都是栈，模拟过程即可。

# 面试题1616

首先想到的方法是单调栈，从左向右遍历建立不严格递增栈，看索引不连续的位置，就是m。同理得到n。

另一种方法：从右向左求最小值的过程中，一边记录右侧的最小值，一边记录比右侧最小值大的元素的位置，最终记录的位置就是m的值。n同理。

# 面试题1626

中缀表达式转后缀表达式（用栈存储运算符，注意优先级，当前大于栈顶则压入，小于等于则弹出直至大于或空）；后缀表达式求值（用栈存储数字，遇到运算符则计算最上面两个数）。

注意数字有可能是多位的，所以不要按字符依次处理。

# 面试题0401

先建立图的邻接表，然后深度优先遍历（不需要回溯，回溯可以求一条路径，此题只要求判断有没有路径）。

# 面试题1705

字母看成1，数字看成-1，算前缀和p[i]，p[i]==p[j]说明i+1到j中字母数字个数相同。

# 面试题1708

此处要求h和w均严格递增：按h递增排序，h相同的按w递减排序（递减排序则可以保证在求LIS时，h相同w不同的人只选一个）；排完序后，求最长递增子序列即可（严格递增的子序列，可以保证w相同h不同的人只选一个，h相同w相同的人只选一个）。

若h和w均不要求严格递增：按h递增排序，h相同的按w递增排序；排完序后，求最长递增子序列即可（非严格递增）。