BITTIGER

CS102 Top100高频算法设计课

第一节　队列、栈和堆

左程云

BITTIGER

版权声明

BITTIGER

Copyright Policy

All content included on the Site or third-party platforms as part of the class, such as text, graphics, logos, button icons, images, audio clips, video clips, live streams, digital downloads, data compilations, and software, is the property of BitTiger or its content suppliers and protected by copyright laws.

Any attempt to redistribute or resell BitTiger content will result in the appropriate legal action being taken.

We thank you in advance for respecting our copyrighted content.For more info see https://www.bittiger.io/termsofuse and https://www.bittiger.io/termsofservice

简单的介绍队列、栈和堆队列

队列：先进先出
栈：先进后出
堆：逻辑结构上是完全二叉树结构，其中每棵子树的最大值节点是头节点。实际结构常使用数组来实现。

常见水题

【题目一】
实现一个特殊的栈，在实现栈的基本功能的基础上，再实现返回栈中最小元素的操作。

【题目二】
怎么用数组结构实现队列和栈结构？

【题目三】
由两个栈组成队列由两个队列组成栈

【总结】
需要的都是小的编程技巧，遇到这种题如果挂了，说明水平不足以拿下offer

如何仅用递归函数和栈操作逆序一个栈

【题目四】
一个栈依次压入1、2、3、4、5，那么从栈顶到栈底分别为5、4、3、2、1。将这个栈转置后，从栈顶到栈底为1、2、3、4、5，也就是实现栈中元素的逆序，但是只能用递归函数来实现，不能用其他数据结构。

如何仅用递归函数和栈操作逆序一个栈　解题思路

【解题点】
1，设计实现一个只删除栈底元素的递归函数记为A
2，利用递归函数A，设计总的逆序递归函数

【关键点】
1，递归函数就是利用函数栈
2，想清楚栈是怎么做的，递归函数就可以怎么做；反之亦然
3，所以一切递归函数都改成非递归的实现

用栈来求解汉诺塔问题

【题目五】
汉诺塔问题比较经典，这里修改一下游戏规则：
不能从最左侧的塔直接移动到最右侧；
不能从最右侧直接移动到最左侧；
最左到最右的互相移动必须经过中间；

求当塔有N层的时候，打印最优移动过程和最优移动总步数。

用栈来求解汉诺塔问题　解题思路

【方法一】
递归的方法。
1，想清楚base case，也就是只剩一层的时候该如何打印
2，剩下所有的过程，统统用递归过程代替

【方法二】
非递归方法
1，明白小压大原则和相邻不可逆原则
2，证明每一步只会有一种操作是达标的，其他的操作都不达标
3，用栈模拟走的过程即可

生成窗口最大值数组

【题目六】

给定一个整型数组arr和整数w，表示一个大小为w的窗口从数组的最左边滑到最右边。求每次窗口内的最大值，并返回结果数组

【举例】

arr = [4, 3, 5, 4, 3, 3, 6, 7]，w = 3时，最终返回[5, 5, 5, 4, 6, 7]

[4 3 5] 4 3 3 6 7	窗口中最大值为5
4 [3 5 4] 3 3 6 7	窗口中最大值为5
4 3 [5 4 3] 3 6 7	窗口中最大值为5
4 3 5 [4 3 3] 6 7	窗口中最大值为4
4 3 5 4 [3 3 6] 7	窗口中最大值为6
4 3 5 4 3 [3 6 7]	窗口中最大值为7

生成窗口最大值数组　解题思路

【解题】
窗口内最大值或最小值的更新结构！这是一种双端队列结构！
以最大值的更新结构为例（最大值的更新结构也类似）
1，对于从右侧窗口新进入的数a，不断从双端队列的队尾释放小于等于a的数，直到遇到一个大于a的数或者双端队列为空时，把a从双端队列的队尾放入
2，对于从左侧窗口新出去的数字b，看看双端队列的头部是不是b（不是说值是不是等于b，而是说是不是b本身）。如果是，把b从双端队列的队列头弹出；如果不是，不进行任何操作
3，双端队列存每个数的下标即可

【总结】
这个结构超级重要！能做很多事！接下来你还会看到！
被火车撞了都不能忘！

构造数组的MaxTree

【题目七】
一个数组的MaxTree定义如下
1，数组必须没有重复元素
2，MaxTree是一棵二叉树，数组的每一个值对应一个二叉树节点
3，包括MaxTree树在内且在其中的每一棵子树上，值最大的节点都是树的头。

给定一个没有重复元素的数组arr，写出生成这个数组的MaxTree的函数，要求如果数组长度为N，则时间复杂度为O(N)、额外空间复杂度为O(N)。

构造数组的MaxTree　解题思路

【方法一】

建立大根堆，注意建立大根堆的过程时间复杂度为O(N)。

【方法二】

找到离一个数最近的、左右两个、比它大或比它小、的两个数

运用“最俗栈结构”！

【总结】
方法二这个结构超级重要！能做很多事！接下来你还会看到！
被火车撞了都不能忘！

出现次数的TOP K问题

【题目八】
给定String类型的数组strArr，再给定整数k，请严格按照排名顺序打印出现次数前k名的字符串。

【题目八扩展】
设计并实现TopKRecord结构。可以不断地向其中加入字符串。并且可以随时打印加入次数最多前k个字符串。

出现次数的TOP K问题　解题思路

【原问题的解题点】
建立大根堆，注意建立大根堆的过程时间复杂度为O(N)。

【扩展问题的解题点】
1，建立hashmap1，收集各种字符串的出现频率
2，建立maxheap，维持当前的TOP K
3，建立hashmap2，维持出现的字符串在堆上的位置
4，完善hashmap1、hashmap2和maxheap之间的配合问题

【总结】
练完拓展问题，你就是堆结构他爹了！父亲节会收到礼物哦～堆爹～

打印所有有序数组中最大的K个数

【题目九】
有N个长度不一样的数组，所有数组中的元素都是从小到大有序排列的，请从大到小打印这N个数组整体的前K个数。

打印所有有序数组中最大的K个数　解题思路

【解题点】
1，每个数组中，从最大的数到最小的数依次成为数组的当前数
2，每个数组的当前数，组成一个大根堆
3，大根堆的堆顶要弹出。弹出是指，选出堆顶的数记为h，h要算作TOPK的一部分；h同时也是所在数组arr的当前数，h弹出后，arr的当前数也要变成下一个
4，进行完步骤3，大根堆的堆顶变成了arr中h这个数的下一个，所以大根堆需要从头部的进行调整，使其重新变成大根堆
5，如果在进行的过程中，某个数组的所有数都已经遍历完，则进行堆的“size–1”操作，即把堆的最后一个数放到堆顶，然后堆的有效区减1，最后从头部的调整堆，使其重新变成大根堆
6，重复步骤3～步骤5，直到选出TOPK为止

【总结】
形象的比喻，一个数组就是一个弹夹，多个弹夹的头部组成的一个堆

随时找到数据流的中位数

【题目十】

有一个源源不断地吐出整数的数据流，假设你有足够的空间来保存吐出的数。请设计一个名叫MedianHolder的结构，可以方便得到吐出所有数的中位数。

随时找到数据流的中位数　解题思路

【解题点】
1，使用两个优先级队列，实质结构就是堆结构
2，大根堆里保存较小的一半；小根堆里保持较大的一半
3，如果大根堆或者小根堆，有一方比另一方多出两个，较多的一方吐出一个数，较小的一方吞下这个数
4，根据大根堆和小根堆的堆顶，可以很方便的得到中位数

【总结】
1，优先级队列的实质结构就是堆结构
2，大根堆和小根堆的联合使用

求最大子矩阵的大小

【题目十一】
给定一个整型矩阵map，其中的值只有0和1两种，求其中全是1的所有矩形区域中，最大的矩形区域为1的数量。
例如：
1 1 1 0
其中，最大的矩形区域有3个1，所以返回3。
再如：
1 0 1 1
1 1 1 1
1 1 1 0
其中，最大的矩形区域有6个1，所以返回6。

求最大子矩阵的大小　解题思路

【解题点】

1，利用逐层累加数组来表示矩阵（以后的课还会见到这个技巧）

2，求某个出发点的扩散区域，其实就是找到离这个点最近的左右两个比它小值的位置

3，利用“最俗栈结构”！来迅速得到某个出发点的扩散区域的大小

4，注意：

等于的情况怎么办？

像小于的情况一样处理！

为什么？

急什么！你们需要的只是等待～

【总结】

飞刀又见飞刀！

最大值减去最小值小于或等于num的子数组数量

【题目十二】
给定数组arr和整数num，返回一共有多少个子数组满足如下情况

Max{arr[i..j]} – Min{arr[i..j]} <= num

其中，Max{arr[i..j]}表示子数组arr[i..j]中的最大值；Min{arr[i..j]}表示子数组arr[i..j]中的最小值

【要求】
如果数组长度为N，请实现时间复杂度为O(N)的解法。

最大值减去最小值小于或等于num的子数组数量　解题思路

【解题点】
1，如果一个子数组不满足条件，那么它怎么阔都不满足
2，如果一个子数组满足条件，那么它怎么缩都满足
3，从位置i开头（初始时令i=0），假设向右边扩到j位置就无法再扩，那么必须以i开头、满足条件的子数组数量为j-1+1，将其加入结果
4，以位置i+1开头，但是j不回退，重复步骤2，直到i走到数组的最后
5，i和j始终组成一个窗口，在这个窗口中，数永远从右边进入窗口，从左边离开窗口，并且我们特别关注这个窗口中最大值和最小值的变化情况……使用窗口内最大值或最小值的更新结构！

IPO

【题目十三】

K代表你一共可以选的项目数，W代表你的初始启动资金，Profits数组代表每个项目的收益， Capital数组代表每个项目需要的启动资金。一个项目能做的条件是你当前的启动资金大于项目的启动资金，一个项目做完后，该项目的收益会加到你的启动资金上。求你能够获得的最大收益。

【要求】

如果两个数组长度为N，请实现时间复杂度为O(N*logN)的解法。

IPO

【解题点】
1，把所有项目加进优先级队列A，并按照所需启动资金升序排列
2，如果当前的启动资金W，可以让一些项目被实现，把这些项目加到优先级队列B，并按照项目收益降序排列
3，如果优先级队列B不为空：在优先级队列B中，选一个受益最大的项目实现，并将该项目收益加到总的启动资金W上。
如果优先级队列B为空：直接返回W，过程结束
4，通过步骤3之后，W已经变大，重复步骤2
5，直到做完K个项目，返回W，过程结束

【总结】

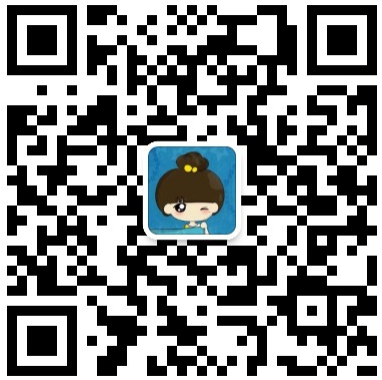
此处应该有666666666666666飞上来！

课程项目负责人：**Catherine**

邮件：wei yi@bittiger.io

左程云答疑邮箱：**chengyunzuo@gmail.com**

微信二维码：



关注微信，获得太阁最新信息

微信: bit_tiger

官网: BitTiger.io