

数据结构(下)

九章算法强化班 第3章



扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: [ninechapter](#)

微博: <http://www.weibo.com/ninechapter>

知乎: <http://zhuoanlan.zhihu.com/jiuzhang>

官网: <http://www.jiuzhang.com>

1. Heap

- Heap基本原理
- Heap 问题的拓展
- Hashheap
- Hashheap 运用

2. Deque

- 双端队列常考一题

Heap

堆

PriorityQueue

Trapping Rain Water

<http://www.lintcode.com/en/problem/trapping-rain-water/>

<http://www.jiuzhang.com/solutions/trapping-rain-water/>

[3,0,1,4,0,1,2]

Trapping Rain Water 2

<http://www.lintcode.com/en/problem/trapping-rain-water-ii/>

<http://www.jiuzhang.com/solutions/trapping-rain-water-ii/>

12	13	8	12
13	4	13	12
13	8	10	12
12	13	12	12

- Key
 - 怎么样通过trapping rain water 1 拓展到这题的思路?
 - 怎么样想到利用堆?
 - 怎么想到由外向内遍历

小技巧

矩阵从外向内遍历技巧



Building Outline

<http://www.lintcode.com/en/problem/building-outline/>

<http://www.jiuzhang.com/solutions/building-outline/>

<https://briangordon.github.io/2014/08/the-skyline-problem.html>

Question ?
堆的原理实现？

- 插入: 将新元素放到 $\text{heap}[\text{size}+1]$ 的位置每次比较它的它父亲元素, 如果小于它的父亲, 证明现在不满足堆的性质, 然后向上Sift Up
- 删除: 将根节点和最后一个节点进行交换如果该节点大于其中一个儿子, 那么将其与其较小的儿子进行交换做Sift Down, 直到该节点的儿子均大于它的值, 或者它的儿子为空
- Key
 - Push – Sift Up
 - Pop – Sift Down
 - Top

```
void siftup(int id) {  
    while (parent(id) > -1) {  
        int parentId = parent(id);  
        if (comparesmall(heap.get(parentId), heap.get(id)) == true) {  
            break;  
        } else {  
            swap(id, parentId);  
        }  
        id = parentId;  
    }  
}
```

```
void siftDown(int id) {
    while (lson(id) < heap.size()) {
        int leftId = lson(id);
        int rightId = rson(id);
        int son;
        if (rightId >= heap.size()
            || (comparesmall(heap.get(leftId), heap.get(rightId)) ==
                true)) {
            son = leftId;
        } else {
            son = rightId;
        }
        if (comparesmall(heap.get(id), heap.get(son)) == true) {
            break;
        } else {
            swap(id, son);
        }
        id = son;
    }
}
```

- How to convert an unordered array into a heap?
- <http://lintcode.com/en/problem/heapify/>
- <http://www.jiuzhang.com/solutions/heapify/>

Break

休息5分钟

HashHeap

- HashHeap

- Key

- Heap + Hash

- 接口

- $O(\log N)$ Push \rightarrow Sift Up
- $O(\log N)$ Pop \rightarrow Sift Down
- $O(1)$ Top
- $O(\log N)$ Delete

- Heap

- 接口

- $O(\log N)$ Push \rightarrow Sift Up
- $O(\log N)$ Pop \rightarrow Sift Down
- $O(1)$ Top
- $O(N)$ Delete

Question?

重复元素怎么办？

Data Stream Median

<http://www.lintcode.com/problem/data-stream-median/>

<http://www.jiuzhang.com/solutions/data-stream-median/>

Sliding Window Median

<http://www.lintcode.com/en/problem/sliding-window-median/>

<http://www.jiuzhang.com/solutions/sliding-window-median/>

- 中位数怎么想到堆
- 窗口操作怎么分解
- How to get idea from the problem that we have solved previously?
 - Such as median and hashheap

小技巧

Sliding Window的题目可以拆解为下面两步

1. 加一个元素
2. 删一个元素

Deque

双端队列

常考题

Sliding Window Maximum

<http://www.lintcode.com/en/problem/sliding-window-maximum/>

<http://www.jiuzhang.com/solutions/sliding-window-maximum/>

[1,2,7,3,8,5,3,2]

- Method 1: for loop $O(nk)$
- Method 2: Balancing Binary Search Tree or Heap: $O(n\log(k))$
 - (a) get max,
 - (b) delete element,
 - (c) insert element
- Method 3: deque $O(n)$
 - (a) pop and push at front,
 - (b) pop at end

Sliding Window Matrix Maximum

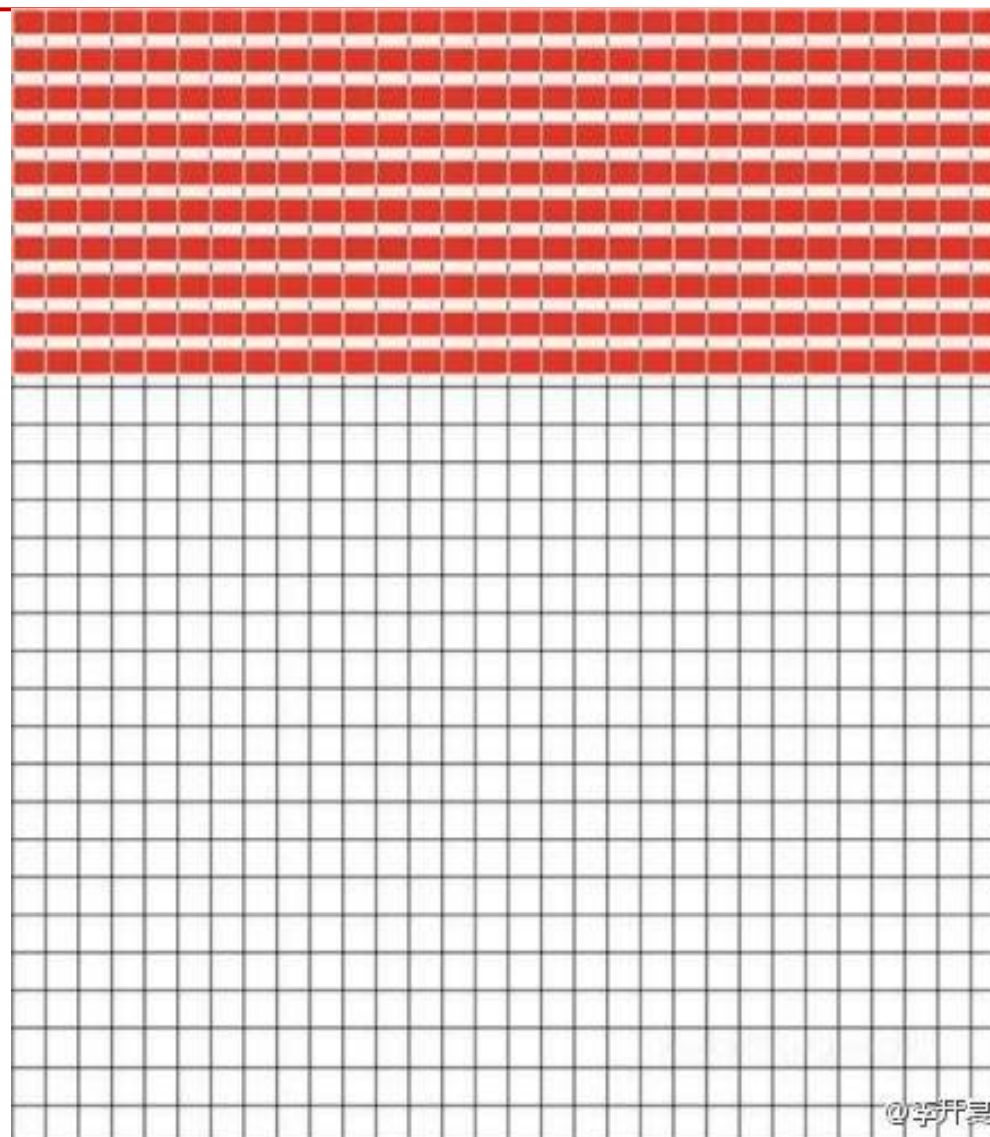
12	8	8	3
13	4	13	12

Sliding Window Matrix Maximum

Sliding Window Maximum
+
Sub Array Sum

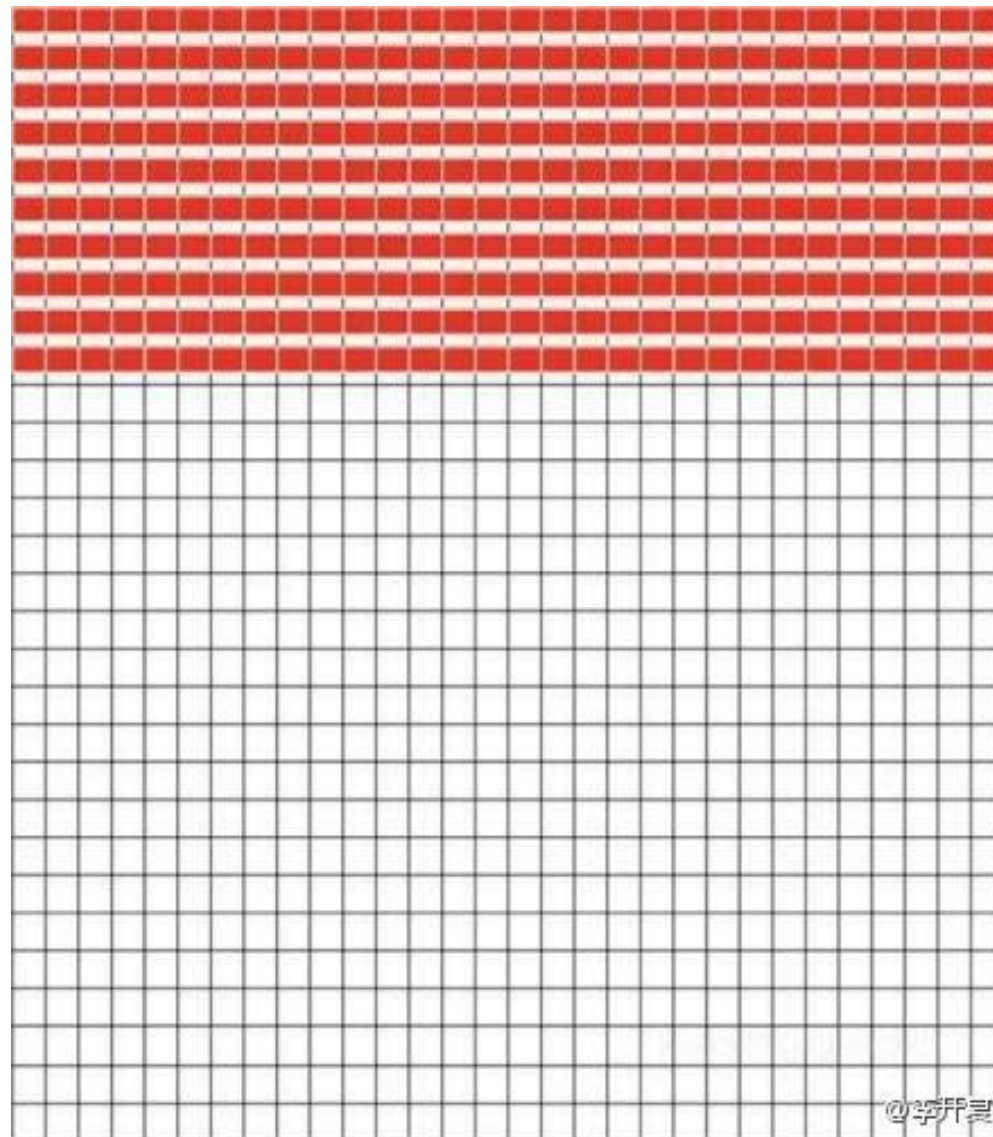
-

- 数据结构的题目：
- Heap: 求集合的最大值
- Hash 带删除操作的Heap
- Deque: 两端都会有push和pop
- Windows problem
 - a.加一个数
 - b.删一个数的方法



@李开复

“人生只有900个月”——事实上，你可以用一张A4纸画一个30×30的表格，每过一个月，就在格子里打勾。你的全部人生就在这张纸上。25岁的人示意如图，暂且珍惜每一天！



Thank You

