

# 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://zhuanlan.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]

### 363. Trapping Rain Water ☆

[Description](#)[Notes](#)[Testcase](#)[Judge](#)

Given  $n$  non-negative integers representing an elevation map where the width of each bar is `1`, compute how much water it is able to trap after raining.



Have you met this question in a real interview?

#### Example

Given `[0,1,0,2,1,0,1,3,2,1,2,1]`, return `6`.

#### Challenge ▾

$O(n)$  time and  $O(1)$  memory

$O(n)$  time and  $O(n)$  memory is also acceptable.

#### Tags ▾

[Array](#)[Two Pointers](#)[Forward-Backward Traversal](#)

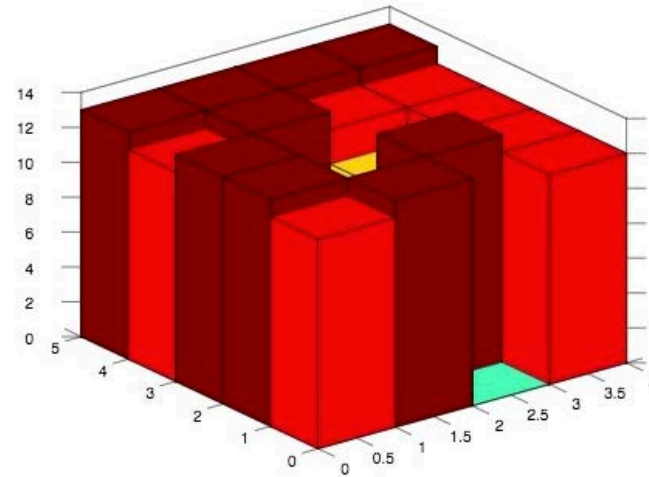
# Trapping Rain Water 2

<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

Given  $n \times m$  non-negative integers representing an elevation map 2d where the area of each cell is  $1 \times 1$ , compute how much water it is able to trap after raining.



Have you met this question in a real interview?

### Example

Given  $5 \times 4$  matrix

```
[12,13,0,12]
[13,4,13,12]
[13,8,10,12]
[12,13,12,12]
[13,13,13,13]
```

return 14.

- 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

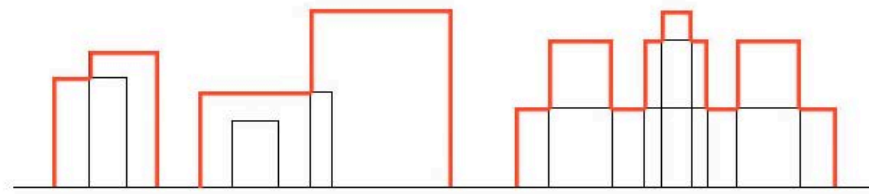
<http://www.lintcode.com/en/problem/building-outline/>

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

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

Given  $N$  buildings in a x-axis, each building is a rectangle and can be represented by a triple (start, end, height), where start is the start position on x-axis, end is the end position on x-axis and height is the height of the building. Buildings may overlap if you see them from far away, find the outline of them.

An outline can be represented by a triple, (start, end, height), where start is the start position on x-axis of the outline, end is the end position on x-axis and height is the height of the outline.



**i Notice**

Please merge the adjacent outlines if they have the same height and make sure different outlines cant overlap on x-axis.

Have you met this question in a real interview?

**Example**

Given 3 buildings:

```
[  
  [1, 3, 3],  
  [2, 4, 4],  
  [5, 6, 1]  
]
```

The outlines are:

```
[  
  [1, 2, 3],  
  [2, 4, 4],  
  [5, 6, 1]  
]
```

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/>

## 130. Heapify ☆

Description

Notes

Testcase

Judge

Given an integer array, heapify it into a min-heap array.

For a heap array A,  $A[0]$  is the root of heap, and for each  $A[i]$ ,  $A[i * 2 + 1]$  is the left child of  $A[i]$  and  $A[i * 2 + 2]$  is the right child of  $A[i]$ .

Have you met this question in a real interview?

### Clarification

#### What is heap?

- Heap is a data structure, which usually have three methods: push, pop and top. where "push" add a new element the heap, "pop" delete the minimum/maximum element in the heap, "top" return the minimum/maximum element.

#### What is heapify?

- Convert an unordered integer array into a heap array. If it is min-heap, for each element  $A[i]$ , we will get  $A[i * 2 + 1] \geq A[i]$  and  $A[i * 2 + 2] \geq A[i]$ .

#### What if there is a lot of solutions?

- Return any of them.

#### Example

Given  $[3, 2, 1, 4, 5]$ , return  $[1, 2, 3, 4, 5]$  or any legal heap array.

#### Challenge ▾

$O(n)$  time complexity

# 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/>

Numbers keep coming, return the median of numbers at every time a new number added.

Have you met this question in a real interview?

### Clarification

What's the definition of Median?

- Median is the number that in the middle of a sorted array. If there are  $n$  numbers in a sorted array  $A$ , the median is  $A[(n - 1) / 2]$ . For example, if  $A=[1,2,3]$ , median is  $2$ . If  $A=[1,19]$ , median is  $1$ .

### Example

For numbers coming list:  $[1, 2, 3, 4, 5]$ , return  $[1, 1, 2, 2, 3]$ .

For numbers coming list:  $[4, 5, 1, 3, 2, 6, 0]$ , return  $[4, 4, 4, 3, 3, 3, 3]$ .

For numbers coming list:  $[2, 20, 100]$ , return  $[2, 2, 20]$ .

### Challenge ▾

Total run time in  $O(n \log n)$ .

### Tags ▾

LintCode Copyright

Heap

Priority Queue

Google



# Sliding Window Median

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

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

Given an array of  $n$  integer, and a moving window(size  $k$ ), move the window at each iteration from the start of the array, find the median of the element inside the window at each moving. (If there are even numbers in the array, return the  $N/2$ -th number after sorting the element in the window. )

Have you met this question in a real interview?

### Example

For array `[1,2,7,8,5]` , moving window size  $k = 3$ . return `[2,7,7]`

At first the window is at the start of the array like this

`[ | 1,2,7 | ,8,5]` , return the median `2` ;

then the window move one step forward.

`[1, | 2,7,8 | ,5]` , return the median `7` ;

then the window move one step forward again.

`[1,2, | 7,8,5 | ]` , return the median `7` ;

### Challenge ▾

$O(n\log(n))$  time

### Tags ▾

LintCode Copyright

Heap

- 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]

## 362. Sliding Window Maximum ★

[Description](#)[Notes](#)[\\_ Testcase](#)[Judge](#)

Given an array of n integer with duplicate number, and a moving window(size k), move the window at each iteration from the start of the array, find the maximum number inside the window at each moving.

Have you met this question in a real interview?

### Example

For array `[1, 2, 7, 7, 8]`, moving window size `k = 3`. return `[7, 7, 8]`

At first the window is at the start of the array like this

`[1, 2, 7]`, return the maximum `7`;

then the window move one step forward.

`[2, 7, 7]`, return the maximum `7`;

then the window move one step forward again.

`[7, 7, 8]`, return the maximum `8`;

### Challenge ▾

$O(n)$  time and  $O(k)$  memory

### Tags ▾

[Two Pointers](#)[LintCode Copyright](#)[Zenefits](#)

- 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

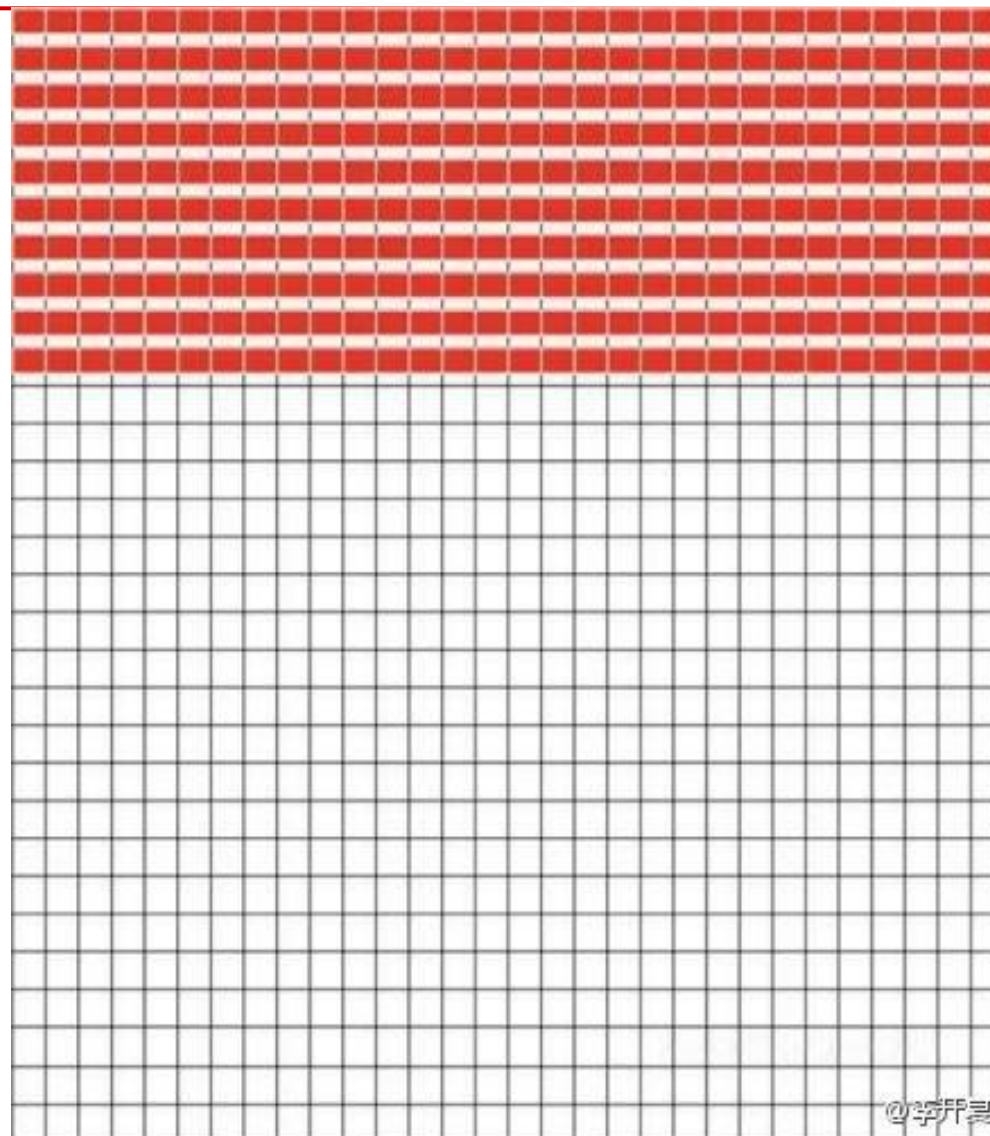
12	8	8	3
13	4	13	12

# Sliding Window Matrix Maximum

Sliding Window Maximum  
+  
Sub Array Sum

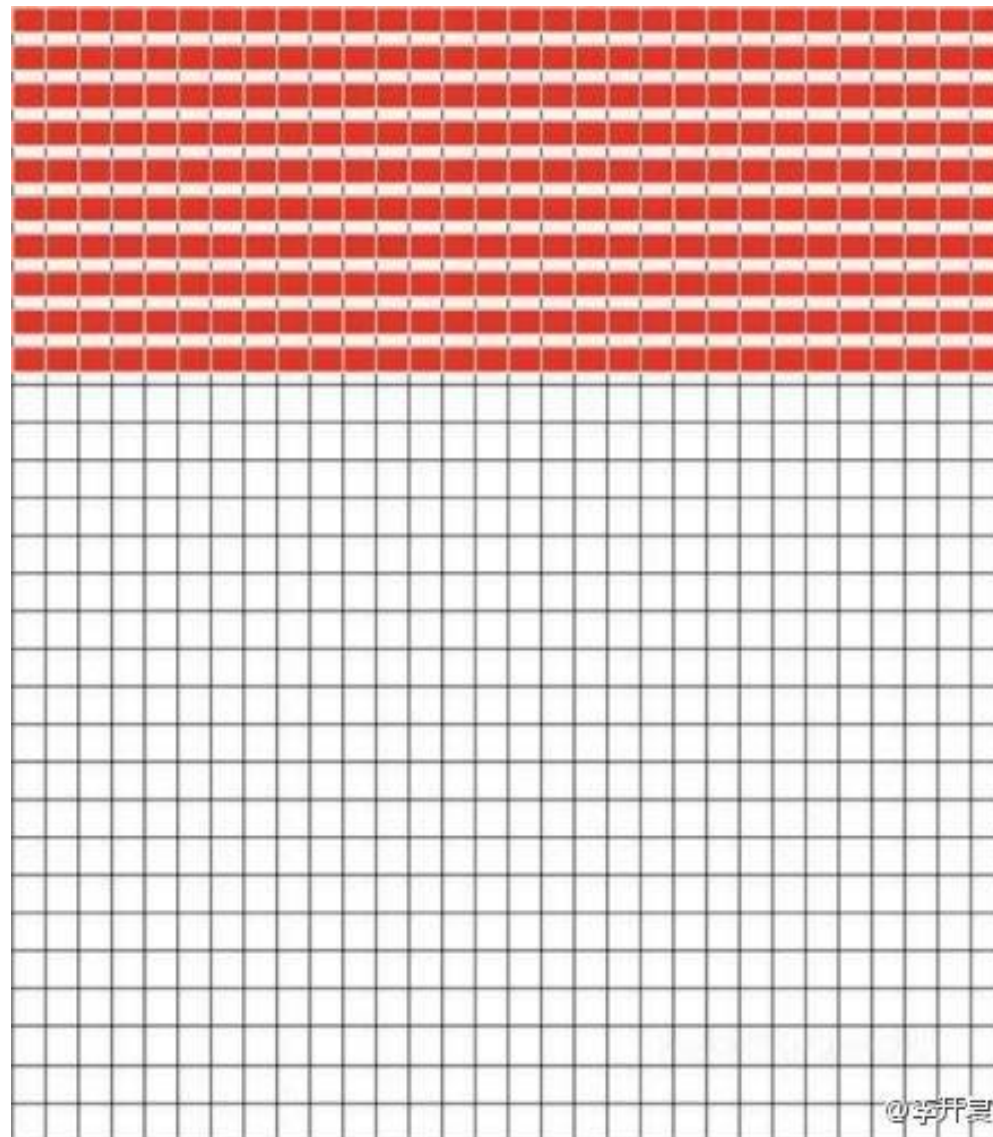
-

- 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

