

# 扫描线算法、双端队列、二分法难题

## Sweep Line, Deque, Binary Search Hard

主讲 侯卫东



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>

# Sweep Line

扫描线算法  
区间问题巧妙解法

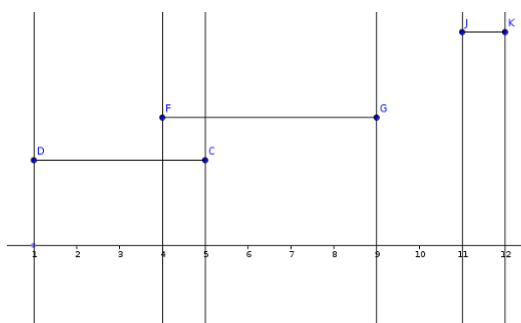
# 扫描问题的特点

1. 事件往往是以区间的形式存在
2. 区间两端代表事件的开始和结束
3. 按照区间起点排序，起点相同的按照终点排序

# 扫描线要点

将起点和终点**打散排序**

$[[1,3], [2,4]] \Rightarrow [[1, \text{start}], [2, \text{start}], [3, \text{end}], [4, \text{end}]]$



# Number of Airplanes in the sky

<https://www.lintcode.com/problem/number-of-airplanes-in-the-sky/>  
<https://www.jiuzhang.com/solutions/number-of-airplanes-in-the-sky/>

## LintCode 391



- 给定n架飞机的起飞降落时间，求最多时天上有多少飞机
- 如果一架飞机的降落时间恰好等于另一架飞机的起飞时间，则认为先降落
- 例子：
- 输入：[[1,10], [2,3], [5,8], [4,7]]
- 输出：3

## 分析：

- 将每架飞机的起降时间作为区间左右端点，建立两个事件
- 对所有事件排序，相同时间的事件降落排在起飞前面
- 扫描线，定义计数器 $C=0$ 
  - 遇到起飞事件， $C+=1$
  - 遇到降落事件， $C-=1$
- $C$ 的最大值即为答案
- FollowUp：如果同时起降，认为先起飞，怎么修改算法

为什么？



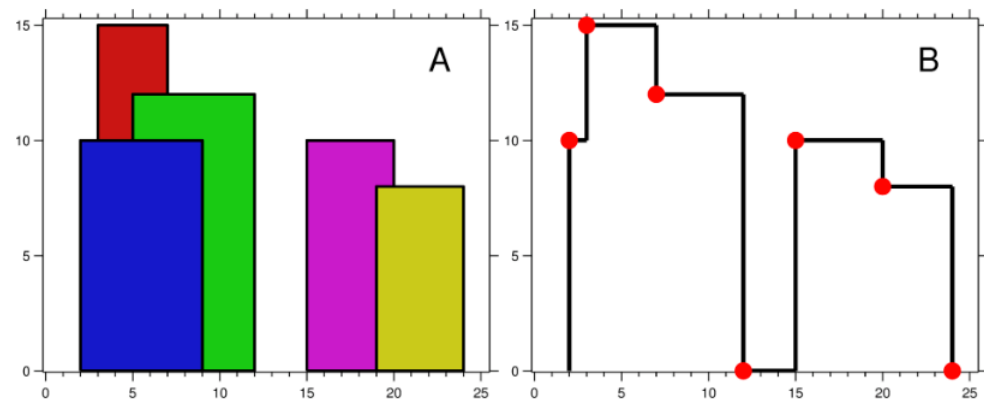


# Building Outline

<https://www.lintcode.com/problem/building-outline/>  
<https://www.jiuzhang.com/solutions/building-outline/>

# LintCode 131

- 给定n个矩形的坐标，底边都在X轴
- 求出所有矩形组成图形的外轮廓线
- 例子：
- 输入：[[2 9 10], [3 7 15], [5 12 12], [15 20 10], [19 24 8]]
- 输出：[[2 10], [3 15], [7 12], [12 0], [15 10], [20 8], [24, 0]]



## 分析：

- 将每个矩形的左边和右边建立两个事件，记下对应高度
- 对所有事件按X坐标排序
- 建立高度的最大堆
- 扫描线
  - 遇到左边事件，堆中加入高度
  - 遇到右边事件，堆中删除高度
- 堆中最大值即为组合图形现在的高度
- 将同一个X坐标的事件全部处理完后，如果新高度 $\neq$ 原来的高度，说明出现拐点，记录下来

## 练习 & 参考资料



类似 Number of Airplanes in the Sky 的一题

<https://www.lintcode.com/problem/meeting-rooms-ii/>

F家高频题：输出两组区间的交集

<https://www.lintcode.com/problem/time-intersection/>

Building Outline 的算法动图模拟

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

# 二分法难题

Binary Search - Hard

# Find Peak Element

<https://www.lintcode.com/problem/find-peak-element/>  
<https://www.jiuzhang.com/solutions/find-peak-element/>

## LintCode 75



- 给定 $A[0..n-1]$ ，其中没有相邻元素相同，并且 $A[0] < A[1]$ ,  $A[n-2] > A[n-1]$
- 找到任意一个 $P$ ，满足 $A[P-1] < A[P] > A[P+1]$
- 例子：
- 输入： $[1, 5, 6, 8, 7, 9, 4]$
- 输出：3

## 分析：

- 首先，这样的P肯定存在
- 因为 $A[0] < A[1]$ ，如果 $A[1]$ 不是要找的元素， $A[1] < A[2]$ ;  $A[2] < A[3]$ ; ...
- 但是 $A[n-2] > A[n-1]$
- 二分查找，对于mid位置，如果 $A[mid] < A[mid+1]$ ，继续向右找；否则向左



# 二分答案

## Binary Search on Result

往往没有给你一个数组让你二分  
同样是找到满足某个条件的最大或者最小值

# 解题方法

通过猜值判断是否满足题意不对去搜索可能解

1. 找到可行解范围
2. 猜答案
3. 检验条件
4. 调整搜索范围

# Sqrt(x)

<https://www.lintcode.com/problem/sqrtx/>

<https://www.jiuzhang.com/solutions/sqrt-x>

**Last** number that  $\text{number}^2 \leq x$

follow up: what if return a double, not an integer?

# Sqrt(x) II

<https://www.lintcode.com/problem/sqrtx-ii/>

<https://www.jiuzhang.com/solution/sqrtx-ii/>

**一直二分直到**  $|\text{number}^2 - x| \leq 1e-10$

# Wood Cut

<https://www.lintcode.com/problem/wood-cut/>

<https://www.jiuzhang.com/solutions/wood-cut/>

## LintCode 183



- 给定n块木头，长度分别是 $L[0]..L[n-1]$
- 要求找到最长的长度s，使得这些木头可以切出至少k块长度为s的木头
- 例子：
- 输入： $L=[232, 124, 456]$ ,  $k=7$
- 输出：114

## 分析：

- 首先，我们发现，对于一个长度 $s$ ，如果可以切出 $t$ 段；而对于另一个长度 $S > s$ ，可以切出 $T$ 段，则一定有 $t \geq T$
- 所以如果长度 $s$ 切出的段数不够 $k$ ，答案肯定比 $s$ 小 → 二分答案！
- 同理，如果长度 $s$ 切出的段数 $\geq k$ ，答案肯定 $\geq s$
- 时间复杂度： $O(n \log L)$





# 课间休息五分钟

---



# Copy Books

<https://www.lintcode.com/problem/copy-books/>  
<https://www.jiuzhang.com/solutions/copy-books/>

- 有N本书需要被抄写，第i本书有A[i]页， $i=0, 1, \dots, N-1$
- 有K个抄写员，每个抄写员可以抄写连续的若干本书（例如：第3~5本书，或者第10本书）
- 每个抄写员的抄写速度都一样：一分钟一页
- 最少需要多少时间抄写完所有的书
  
- 例子：
- 输入：
  - $A = [3, 2, 4]$ ,  $K=2$
- 输出：
  - 5（第一个抄写员抄写第1本和第2本书，第二个抄写员抄写第3本书）

## 分析：

- 题目要求K个抄写员抄完最少需要的时间
- 反过来想，如果我们限定时间不超过T，最少需要的抄写员
- 这个问题比较好做，贪心法
  - 从第一本书开始，第一个人一直抄到时间即将超过T
  - 第二个人，...
- 如果需要的抄写员 $>K$ ，说明答案一定 $>T$ ；反之答案 $\leq T \rightarrow$ 二分答案！

# Find The Duplicate Number

<https://www.lintcode.com/problem/find-the-duplicate-number/>  
<https://www.jiuzhang.com/solutions/find-the-duplicate-number/>

## LintCode 633

- 给定一个长度为 $n+1$ 的数组，其中均为1到 $n$ 之间的整数
  - 保证只有一个数字重复了多次
  - 找到这个数字
  - 辅助空间只能 $O(1)$
- 
- 例子：
  - 输入：
    - $[5,5,5,1,2,3]$
  - 输出：
    - 5

## 分析：

- 假设答案是S
- 数组一定是（假设排好序）： $[1, 3, \dots, S, \dots, S, S+1, \dots, n]$
- 那么其中 $\leq S$ 的数大于S
- 而且对于所有 $T \geq S$ ,  $\leq T$ 的个数大于T
- 而对于所有 $T < S$ ,  $\leq T$ 的个数小于等于T
- 二分法,  $O(n \log_2 n)$

# Maximum Average Subarray II

<https://www.lintcode.com/problem/maximum-average-subarray-ii/>  
<https://www.jiuzhang.com/solutions/maximum-average-subarray-ii/>



## LintCode 617



- 给定一个数组A，找到其中平均值最大的子数组，要求长度 $\geq k$
- 例子：
- 输入：
  - [1, 12, -5, -6, 50, 3],  $k = 3$
- 输出：
  - 15.667 //  $(-6 + 50 + 3) / 3 = 15.667$

## 分析：

- 如果要求和最大，可以用前缀和数组。但是平均值最大不好求
- 那么如果最大平均值是 $T$ ，那么我们的目标是找到
  - $(A[\text{left}] + \dots + A[\text{right}]) / (\text{right} - \text{left} + 1) \geq T$ ，且  $\text{right} - \text{left} + 1 \geq k$
  - 即  $(A[\text{left}] - T) + \dots + (A[\text{right}] - T) \geq 0$
- 换句话说，对于一个 $T$ ，把每个元素 $A[i]$ 减去 $T$ 得到 $B[i]$
- 希望找到最大的 $B[\text{left}] + \dots + B[\text{right}] \geq 0$ ，且  $\text{right} - \text{left} + 1 \geq k$
- 这可以通过前缀和实现
- 如果找不到这样的 $(\text{left}, \text{right})$ ，说明答案小于 $T \rightarrow$ 二分答案



# Deque

双端队列  
两端都会有push和pop

## 常考题

# Sliding Window Maximum

<https://www.lintcode.com/problem/sliding-window-maximum/>  
<https://www.jiuzhang.com/solutions/sliding-window-maximum/>

## LintCode 362

- 给定一个数组A，找到其中每个大小为k的窗口中最大值
- 例子：
- 输入：
  - [1, 2, 7, 7, 8], k = 3
- 输出：
  - [7, 7, 8]

## 分析：

- 类似单调栈，但是两端都有操作
- 基本思想：如果 $A[i] \leq A[j]$ ，且 $i < j$ ，那么 $A[i]$ 就没有用了，即以后永远不会成为窗口最大值
- 窗口向右移动，左端元素移出队首（如果仍在队列中），右端元素 $A[j]$ 移进队尾，并删除所有 $\leq A[j]$ 的 $A[i]$
- 时间复杂度 $O(N)$





## 今日经典三题

---



- Number of Airplanes in the sky
  - 扫描线经典入门题目
- Maximum Average Subarray II
  - 二分答案的典型题目
- Sliding Window Maximum
  - 滑动窗口经典题型

## 思路总结

- 扫描线
  - 见到区间需要排序就可以考虑扫描线
- 二分法
  - 按值二分，找到单调的地方
- 双端队列
  - 维护一个候选可能的最大值集合

# [推荐]Debug 的基本步骤

<http://www.jiuzhang.com/qa/3815/>