

数据结构1st

XJTU

Information and Computational Science

mg

xjtumg.me

xjtumg1007@gmail.com

- 单调栈
- 单调队列
- 优先队列

- POJ 2796
- 给定 n 个非负整数
- 一个区间的权值定义为这个区间的最小数乘以这个区间所有数的和
- 求最大权值的区间
- $n \leq 1e5$
- $A_i \leq 1e6$

- 转化问题
- 枚举最小值，“拓展”区间

- ST表预处理
- 枚举最小值，二分区间长度
- $O(n \log n)$

- 利用单调栈(栈顶最大)
- 假设单调栈内元素为 $S_1 \dots S_k$
- 则 $S_i \sim S_{i+1}$ 之间的所有数均大于 S_{i+1}

- 对于新进栈的元素 A_i
- `while (!S.empty() && S.top() >= A_i)`
- `S.pop();`
- 此时`S.top()`是 A_i 左边第一个比 A_i 小的数
- 可以把S中的每个数想象成一堵墙
- 例子：3 1 6 4 5 2

- 两个方向分别算一次
- 每个点最多进队一次出队一次
- $O(n)$

- 求栅栏(柱形图)的最大子矩形

- $n \leq 1e5$

- POJ 2823
- 给你一个长度为N的数组，一个长为K滑动的窗体从最左移至最右端，你只能见到窗口的K个数，每次窗体向右移动一位，如下表：
- [1 3 -1] -3 5 3 6 7 -1 3
- 1 [3 -1 -3] 5 3 6 7 -3 3
- 1 3 [-1 -3 5] 3 6 7 -3 5
- 1 3 -1 [-3 5 3] 6 7 -3 5
- 1 3 -1 -3 [5 3 6] 7 3 6
- 1 3 -1 -3 5 [3 6 7] 3 7
- $n \leq 1e6$

- ST表预处理

- $O(n \log n)$

- 线段树在线处理
- $O(n \log n)$
- 为什么不能用树状数组？

- 以最大值为例
- 建立一个单调递减队列(队头最大)
- 元素从左到右依次入队，入队之前必须从队列尾部开始删除那些比当前入队元素小或者相等的元素，直到遇到一个比当前入队元素大的元素，或者队列为空为止
- 若此时队列的大小超过窗口值，则从队头删除元素，直到队列大小小于窗口值为止。然后把当前元素插入队尾
- 当前区间的最大值即为队头

- 每个元素最多只进队一次出队一次
- $O(n)$

- POJ 3017
- 将一个由N个非负整数组成的序列划分成若干段，要求每段数字的和不超过M，求每段的最大值的和最小的划分方法，输出这个最小的和
- $n \leq 1e5$

- $dp[i] = \min(dp[j] + \max\{A[j + 1] \dots A[i]\})$
- $j < i$
- $\text{Sum}\{A[j + 1] \dots A[i]\} \leq M$
- $O(n^2)$

- 假设 x 为 $A[j + 1] \dots A[i]$ 最大值的位置，即
- $A[x] = \max\{A[j + 1] \dots A[i]\}$
- 则，对于任意 y 满足 $y \geq j \ \&\& \ y < x$
- $\max\{A[y + 1] \dots A[i]\}$ 为定值
- 又， $A[i]$ 非负，则 $dp[i]$ 单调不减
- 综上，在求 $dp[i]$ 时，对于最大值相等的区间应取最前面的

- 联想到单调队列
- 维护一个递减的单调队列(队头最大)
- 则队列内的每个元素即为对应最大值的区间最左边的位置
- 设队列为 $Q_{head} \sim Q_{tail}$ ，则对于 Q_i ，前面第一个比它大的数为 Q_{i-1}
- 即，如果 $\max\{A[j+1] \dots A[i]\} = Q_i$ 则 j 取值范围为 $Q_{i-1} \sim Q_i - 1$
- 则取最左边的位置 Q_{i-1}

- 队尾已经可以合理维护了
- 如果队头到当前点的所有元素和 $>M$ 则弹出队头
- 那么现在单调队列内包含了所有可能会更新答案的点
- 对于在单调队列内的点暴力更新答案
- $O(n^2)$ 数据弱可以AC

- `multiset`维护单调队列内的答案即可
- $O(n \log n)$

- POJ 1442
- 序列上两种操作
 - 1. 加一个数
 - 2. 求整个序列的第 i 小数
- 第一次操作2的时候 $i = 1$ ，第二次 $i = 2$ ，以此类推
- $n \leq 60000$

- 两个优先队列
- 保存前k小的数，大数在前
- 保存后n - k小的数，小数在前
- 维护即可
- $O(n \log n)$

- NYOJ Interesting Punch-Bow!
- $n * m$ 的矩阵，每个位置的数字代表高度(可以想象成 $n * m$ 个柱子)
- 问最多能蓄多少水？想象拿大量水灌
- 边缘旁边是空的不能蓄水
- $n, m \leq 300$

- 用优先队列维护“边界”
- 每次取高度最低的扩边界
- $O(n * m * \log(n * m))$