序列杂题选讲

课后练习

- 给定n个区间[ai,bi]，这些区间的并可以表示为一些不相交的闭区间的并。你的任务就是在这些表示方式中找出包含最少区间的方案。（你的区间不一定从给定区间中选）
- 5
- 5 6
- 1 4
- 10 10
- 6 9
- 8 10
- [1,4],[5,10]

- 按左端点排序。
- 记录当前答案区间的左右端点 $l = a_1, r = b_1$ 。
- 若 $a_{i+1} \leq r$ 则相交，更新 $r = b_{i+1}$ ，左端点不变。
- 否则不相交，输出旧的答案区间，
- 新的答案区间 $l = b_{i+1}, r = b_{i+1}$ 。

- 给定n和n个区间[ai,bi]
- 给定m，m个人，每个人有一个区间[ci,di]和一个ki。
- 如果一个区间[ai,bi]被一个人的区间[cj,dj]包含，则这个人可以选择这个区间。
- 每个区间至多被一个人选择，每个人至多选ki个区间。
- 求是否所有区间都能被某个人选中。
- [1,3],[2,4],[3,5]
- [1,4],2
- [2,5],1

- 对于左端点为1的区间，它们只能被左端点为1的人选择。
- 左端点为2的区间，能被左端点为1或2的人选择。
- 将所有人和区间按左端点排序，相同时把人排在前面。
- 按顺序枚举，若枚举到了区间，则需要在它之前的人里选出一个 $d_j \ge b_i$ 的人，同时 d_j 应尽量小。
- 对人维护一个set，枚举到人时加入pair(di,ki)，枚举到区间时查找最小的 d_j ，且 $d_j \ge b_i$ ，让他的k--，若k=0则移出set。

课后练习

- 给定n和n个数ai。
- 求长度最大的区间[l,r]，满足：
- a[l]是该区间中的最小值（没有之一）
- a[r]是该区间中的最大值（没有之一）
- 5
- 1 2 3 4 4

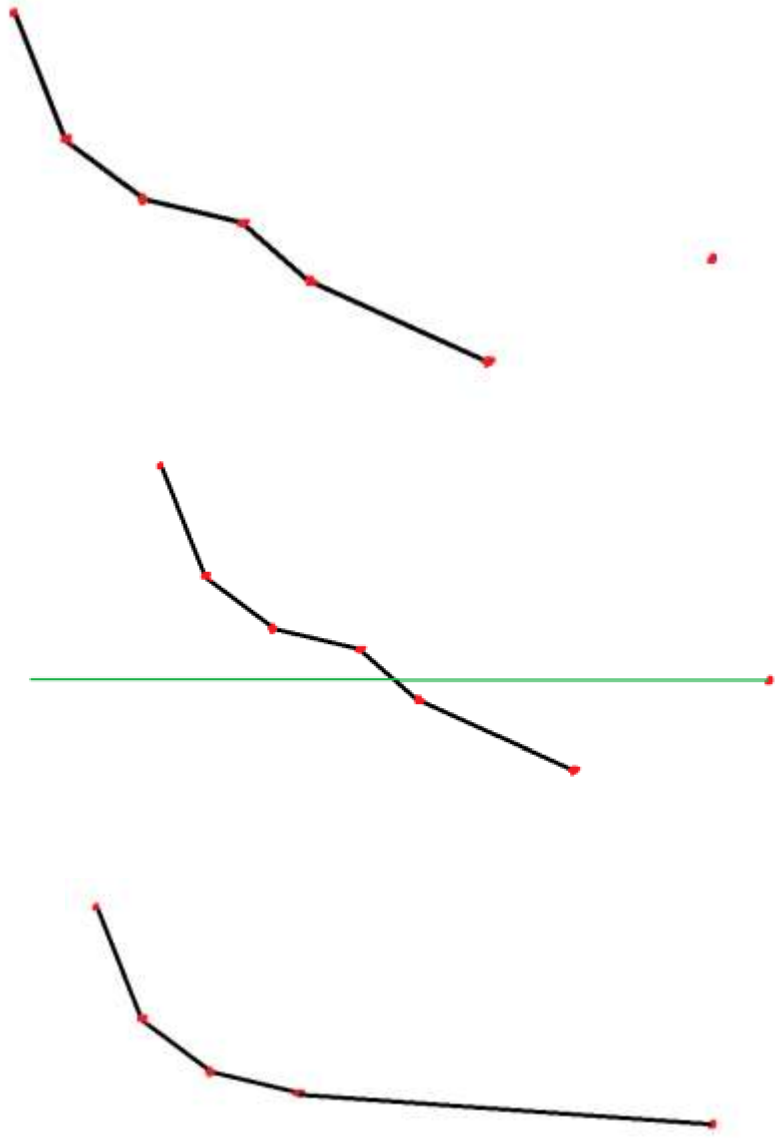
- [1,4]

- 左端点l满足： $a_l = \min(a_l, a_{l+1}, \dots, a_r)$ 。
- 从左往右枚举右端点r，记录满足上述条件的所有左端点。
- 从小到大列为 $b_1, b_2, \dots, b_k$ 。
- 对于下一个元素 a_r ，若 $a_r \le b_k$ ，则 b_k 不再满足前述条件。
- 若 $a_r \le b_{k-1}$ ，则 b_{k-1} 不再满足前述条件。
- 调整b序列，直至 $a_r > b_i$ ，将 $b_{i+1} = a_r$ 。
- 右端点满足条件?
- 设在r之前最近的>=a[r]的位置为x，则左端点大于x即可。
- 在b序列中二分x的位置。
- 如何找到x?
- 类似维护b序列，换一个方向，维护一个c序列。

单调栈

- {3,4,2,5,1}

- {3}
- {3,4}
- {2}
- {2,5}
- {1}



- 给定n和长为n的序列ai。
- 对于一个x，选择ai的代价为 $|a_i - x|$ 。
- 给定代价之和的上限B，求出一个x，使得选择的ai最多。

- 5 6
- 1 2 10 12 14

- x=12，选择10，12，14

- 对ai排序后，选择的ai一定是在一个区间[l,r]内的。
- 若确定了区间，代价之和为 $f(l,r) = \sum_{i=l}^{r} |a_i - x|$ 。
- 要使其最小，x一定选在这些ai的中位数。
- 假设对于当前左端点L，求出了最大的R满足f(L,R)<=B。
- 在L变大的情况下，对应的满足f(L,R)<=B的最大的R不会减小。
- 枚举L，维护最大的满足f(L,R)<=B的R。
- two pointers

- 给定n和n个正整数ai。
- 至多m次操作，每次可以把一个大于0的ai减1。
- 你的目标是，存在某个ak=0的前提下，最小化 $\max(|a_i - a_{i+1}|)$

- 16 15
- 8 7 6 5 5 5 5 5 6 6 7 8 9 7 5 5

- 0 2 4 5 5 5 5 5 6 6 7 8 9 7 5 5

- 二分答案x。
- 没有ak=0的限制?
- 只能减少ai。
- 从左往右， $a_i = \min(a_i, a_{i-1} + x)$ 。
- 从右往左， $a_i = \min(a_i, a_{i+1} + x)$ 。
- 某位置ak为0?
- p*x,(p-1)*x,…,x,0,x,2*x,…
- 对于a[i]>(k-i)*x，a[i]=(k-i)*x（i<k)
- 对于a[i]>(i-k)*x，a[i]=(i-k)*x（i>k)
- 随着k的递增，满足条件的区间左右端点都递增。
- two pointers

课后练习

- 给定n个n个区间[ai,bi]，和一个m。
- 你需要选择m个区间，在保证它们至少有一个公共点的前提下，最小化选出的区间长度的极差。

- ai<=x<=bi
- max(bi-ai)-min(bi-ai)

- 选m个区间包含共同点等价于选若干区间，每个区间内所有位置+1，存在某个位置大于等于m。
- 按长度对区间排序，在长度最小值为L的前提下，向右枚举R，直到某位置被覆盖至少m次。
- L递增时，R单调不降。
- two pointers + 线段树维护区间和、最大值

- 给定n，一个仅含a和b的长为n的字符串，和一个k。
- 可以改变至多k个字符，求改变后最大的连续相同字符的数量。

- 8 1
- aabaabaa

- 5

- 二分答案x。
- 枚举所有长度为x的区间，前缀和求得a数量和b数量，看较小的那个与k的关系。

- 给定n和长为n的序列a。
- 定义f[l,r]为区间[l,r]内的不同元素数量。
- 随机选取l和r，求f[l,r]的期望。

- 1 1 2
- f[1,3]=2

- 对于区间内的相同元素，只会产生一次贡献。
- 把它记在区间内最靠左的该元素上。
- 对于任一元素ai，记前一个与它相同的元素位置是last，则它能产生贡献的区间左端点为last+1到i，右端点为i到n。

课后练习

- 数轴上有n枚不能动的棋子，位置分别为ai。
- 你有一枚可移动的棋子，设其当前位置为x，你可以将其移动到y，当：
- 1.恰有一枚不能动的棋子在x与y之间
- 2.该不能动的棋子到x和到y的距离相等
- 3.y<x
- 类似于跳棋，只能隔着一个棋子往左跳。
- 多次询问，每次给定初始位置，求最多能跳几次。

```
3 3
3 5 8
4 6 7
```

```
1
2
0
```

- 考虑两枚相邻的不动棋子 a_i, a_{i+1} 之间的区域。
- 最大跳跃次数相同的位置一定组成连续区间。

- 假设维护了该结构，现在考虑 a_{i+2} ，下一枚不动棋子。
- 显然会先进行一次翻转。
- 若 $a_{i+2}-a_{i+1}<a_{i+1}-a_i$ ，则会删去结尾若干段。
- 反之，则会在结尾加上跳跃次数为0的一段。
- 1.不计跳跃次数，而记最终跳跃到的目标（在哪两个棋子之间）。
- 2.每一段记录长度。
- 离线询问，在枚举到对应的区间时处理，找到询问位置所在的段。

- 如何查找?
- 如果能维护出前缀和，就可以二分了。
- 如何维护该结构?
- 不考虑翻转，而是在两端进行操作。
- 1.前端插入，后端插入
- 2.前端删除，后端删除
- 3.求前缀和，求后缀和
- 只记前缀和以及总和。
- 对于后端的插入和删除，正常维护前缀和。
- 对于前端的插入和删除，相当于对所有前缀和产生影响。
- 用一个懒标记存储。

课后练习

- 给定n和长为n的序列a。
- 求其严格上升子序列（可以不连续，至少2个数）的个数。
- 相同的子序列只算一次。

- 4
- 1 2 3 3

- {1,2}{1,3}{2,3}{1,2,3}

- 不考虑重复?
- f[i]表示以ai为结尾的上升子序列个数。
- $f[i] = \sum_{j<i,a[j]<a[i]} f[j]$
- 离散化后，树状数组优化。
- 如何考虑重复?
- 若之前已经计算过以数值c结尾的答案，当前ai也是c，那么前述计算方法会将之前的答案再加一遍。
- 记录每个数值上一次作为结尾的答案。
- 先按前述方法计算，减去该数值上一次的答案后，就是这一次产生的新的子序列个数。

- n瓶牛奶，每瓶牛奶有一个保质期ai，ai天之后就会过期。
- 每天会喝k瓶，具体地，每天会选择最接近过期的k瓶。
- 如果某天有牛奶过期，就失败了。
- 在第一天开始之前有一次购物，商店里有m瓶牛奶，每瓶牛奶的保质期为bi。
- 你需要买尽量多的牛奶，同时要保证喝牛奶计划的成功。
- 3 6 2
- 1 0 1
- 2 0 2 0 0 2
- 3
- 1 2 3

- 对于所有的t，保质期不超过第t天的牛奶总数小于等于t*k，计划就是成功的。
- 让所有牛奶尽可能晚地喝。
- 按保质期从大到小考虑商店里的牛奶bi。
- 若当前<=bi的牛奶总数小于bi*k，则可以选择。
- 反例?
- 4 1 2
- 0 1 1 1
- 0
- 要考虑到所有>bi的天数，每一天都有一个限制
- $tmp = \min\left(tmp, b_i \times k - num_{\leq b_i}\right)$
- 选择之后，tmp--

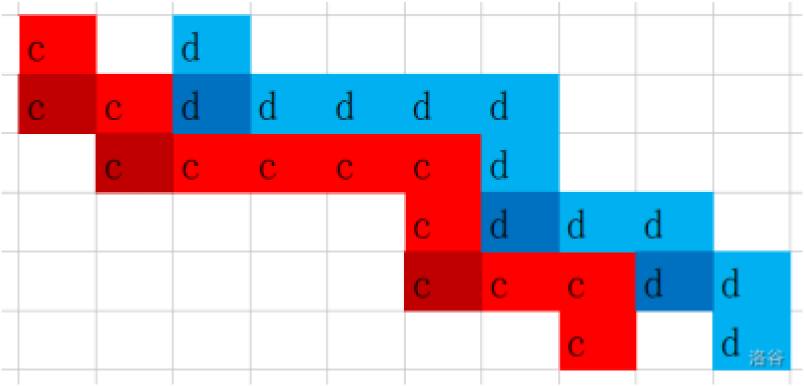
课后练习

- 给定长为n的字符串s和t。
- 一次操作s->s’：
- 从左往右，对于每个s’[i]，可以设为s[i]，也可以设为s’[i-1]。
- 目标：最少的操作次数将s转化为t。
- abcde
- aaacc
- abcde->aaccc->aaacc

- 一段连续字符可在一轮之内完成。
- t中的字符tj，从s中的si传播过来（i<j且s[i]=t[j]）
- 什么情况下需要多次操作?
- 右侧被需要传播的字符阻挡了。
- 对于最右侧的传播，
- 第一轮就传到最靠右侧。
- 每一个传播过程都应尽量靠右。

S0	a		b	c		a	b					
S1	a		b	c	c	a	b	b	b	b	b	b
S2	a		b	b	c	a	a	a	a			b
S3	a			b	c	c	c	c	a			b
S4	a	a	a	b	b	b	b	c	a	a	a	b

- 从后往前维护传播路径的转折点序列。
- 对于下一条传播路径si->tj，
- 根据j来删除末尾若干个转折点，
- 根据i来增加开头的一个转折点。



- 给定n和长为n的序列a。
- 一次操作：将a的某个元素+1或-1。
- 求：最少几次操作，使得a严格上升。

- 5
- 5 4 3 2 1

- 12
- 1 2 3 4 5

- 如果是不严格上升？ （即可以相等）
- 最终a序列的每个元素一定都是初始a序列的某个元素。
- f[i][j]：前i个元素不严格上升，最后一个元素被调整为初始序列第j大的元素，所需的最小操作次数。
- f[i][j]=f[i-1][k]+abs(a[i]-b[j])（k<j）
- 记一下前缀min
- 考虑严格上升?
- 令a[i]=a[i]-i，就转化为了不严格上升。

Slope Trick

- 分段线性凸函数：

$$f(x)=\begin{cases}-x-3 & (x\le -1)\\ x & (-1<x\le 1)\\ 2x-1 & (x>1)\end{cases}$$

- 只记录最右边的一段f(x)=2x-1，以及断点集合。
- 断点出现次数为斜率在该点上改动的大小。
- {-1,-1,1}
- 函数相加?
- 最右端函数相加，断点集合合并。

Slope Trick

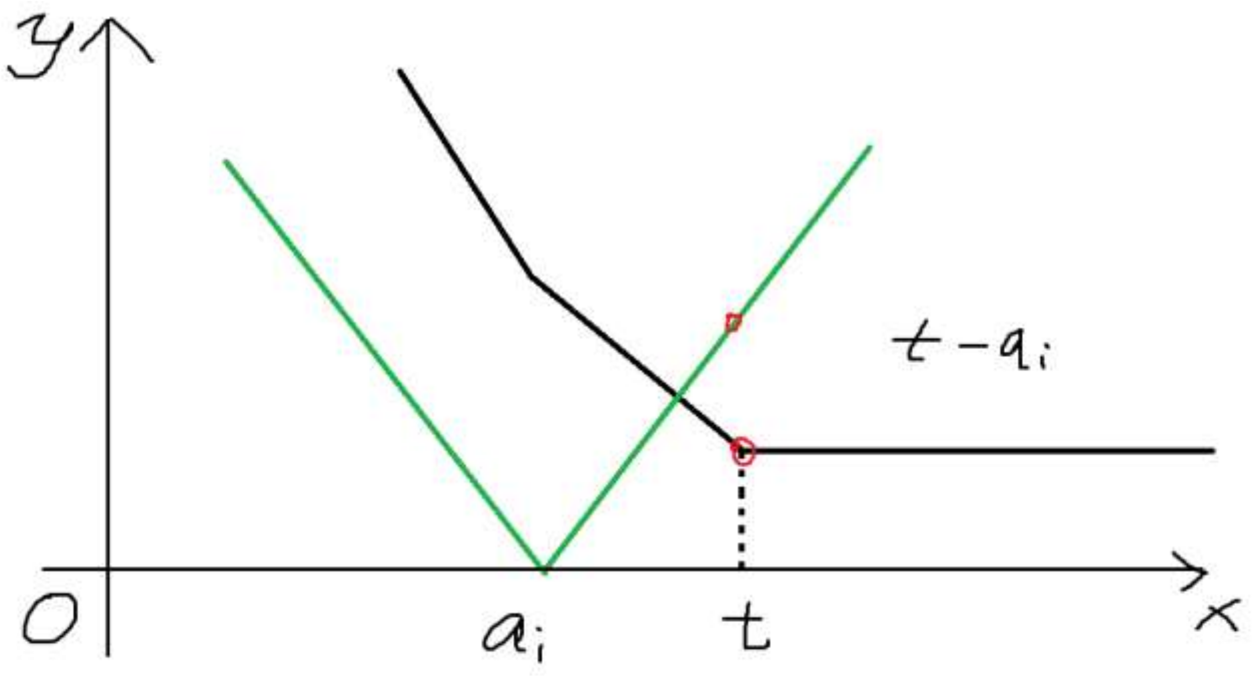
- 以j结尾：
- $f[i][j] = \min\limits_{k \le j}(f[i-1][k]) + |a[i] - j|$
- 记$F_i(x) = f[i][x]$，$G_i(x) = \min\limits_{k \le x}\left(F_{i-1}(k)\right)$
- $F_i(x) = G_i(x) + |a[i] - x|$
- 绝对值为分段线性凸函数，取min能保证该性质。
- $F_i(x), G_i(x)$均为分段线性凸函数。

Slope Trick

- 本题中求最小值，对于凸函数，也就是斜率为0处。
- $|x-a_i| = \begin{matrix} a_i - x, x \le a_i \\ x - a_i, x > a_i \end{matrix}$
- 一个断点，斜率改变2，断点集合为 $\{a_i, a_i\}$ 。
- 维护斜率<=0的断点，每次加入两个断点后，最后一段的斜率为1，因此需删去最大的断点。
- {…,-3,-2,-1,0}->{…,-3,-2,-1,0,1}

Slope Trick

- 右端函数?
- 斜率为0，就是常数。



- 用大根堆维护断点序列。
- 每次加入两个 a_i ，弹出最大值 t ， $ans += t - a_i$ 。

- 对于一个排列p和正整数k，定义f(p,k)=
- $p_2, p_3, \dots, p_k, p_1$ ，$p_{k+2}, p_{k+3}, \dots, p_{2k}, p_{k+1}$ ，$\dots$ ，$p_{rk+2}, \dots, p_n, p_{rk+1}$
- 也就是每个长为k的部分向前循环移一位（最后一部分可能不足k）
- 给定n，p初始为{1,2,…,n}
- 求f(f(…f(f(p,2),3)…,n-1),n)

- 4
- f({1,2,3,4},2)={2,1,4,3}
- f({2,1,4,3},3)={1,4,2,3}
- f({1,4,2,3},4)={4,2,3,1}

- $p_2, p_3, \ldots, p_k, p_1$, $p_{k+2}, p_{k+3}, \ldots, p_{2k}, p_{k+1}$, $\ldots, p_{rk+2}, \ldots, p_n, p_{rk+1}$
- 观察该式， p_1 移到了原本 p_{k+1} 的位置
- p_{k+1} 移到了原本 p_{2k+1} 的位置
- 所以只要改动这些位置即可，在序列头删去一个元素，序列尾添加一个元素。
- $\sum_{i=2}^{n} \frac{n}{i} = O(nlogn)$

课后练习

- 给定n和长为n的序列a，一次操作为：
- 选择一个区间[l,r]，将区间内的所有数替换为该区间的中位数。
- 操作不超过k次，要使得最终a的最小值最大。
- 长度为m的序列的中位数定义为第 $\left\lceil \frac{m}{2} \right\rceil$ 小的数。

- 第一步显然是二分答案转化为01序列。
- 设0的权值为-1，1的权值为1，则一次操作就是把一个权值和为正的区间设为1。
- 每次操作的区间一定会尽可能长，在不取[1,n]的情况下，权值和一定是1，也就是1恰好比0多一个，因此每次操作减少的0会翻倍，最多操作O(logn)次。
- 对于操作的区间序列，后一个区间一定包含前一个区间，最后一个区间是[1,n]。
- f[i][l][r]：第i次操作，能否将区间[l,r]全部变为1。
- f[i][l]：第i次操作，以l为左端点，最大的能全变为1的右端点。

- 设$g[l][r]$表示l到r中0的数量乘以2，也就是全部变1后权值的变化。
- 若第i次想要操作$[l,r]$，第$i-1$次操作了$[p,q]$，那么在l到r的权值和再加上$g[p][q]>0$时可行。
- 设权值前缀和为$s[i]$，也就是$s[r]-s[l-1]+g[p][q]>0$，$s[l-1]-g[p][q]<s[r]$。
- 从后往前考虑每个区间，更靠右的，且贡献更小的区间不再有用。
- 第i次操作合法的右端点r?
- $r>=f[i-1][p]$且$s[r]>s[l-1]+g[p][q]$
- 记$h(k)$为$s[r]>k$的最大r
- $h(s[l-1]+g[p][q])>=f[i-1][p]$
- h单调，$s[l-1]$单调，$g[p][q]$单调，$f[i-1][p]$单调