知识精炼（四）

主讲人：邓哲也

Codeforces 605D

给一个长度为 n 的序列 A[1], A[2], … ,A[n].

每次在原序列上做一个操作，把A[i]修改为 k。

问每次操作后，序列的最长严格上升子序列的长度。

n, m <= 400000

input

```
4 4
1 2 3 4
1 1
1 4
4 3
4 5
```

output

```
4
3
3
4
```

Codeforces 605D

回忆一下最长上升子序列（LIS）是怎么做的：

f[i] 表示以 i 结尾的最长递增子序列的长度。

f[i] = max{ f[j]+1 | 1 <= j < i, A[j] < A[i] }

暴力 $O(n^2)$

Codeforces 605D

我们用一个栈 stack 来存 min{ A[k] | f[k] = i }

显然这个栈是单调的。

计算 f[i] 时，在栈中找到一个最大的 j 满足 stack[j] < A[i]

那么 f[i] = j + 1。

然后用 A[i] 去替换 stack[j + 1]。

最后答案就是 max { f[i] }。

Codeforces 605D

回到这道题目。

对于修改 A[i] = x。

如何求一定包含 A[i] 的 LIS 呢?

假设我们知道了以第 i 个数结尾的 LIS —— left[i]

和从第 i 个数开头的 LIS —— right[i]

我们只要求出 max{ left[k] | k < i, A[k] < x }

和 max{ right[k] | k > i, A[k] > x }

两者相加再加上 1（x自己），就是包含 x 的 LIS!

Codeforces 605D

因此我们需要先计算 left 和 right 数组。

用之前提到的方法，f 就是 left 数组。

同理，倒着做一遍，可以得到 right 数组。

这样包含 x 的 LIS 就被我们求出来了。

如果这个值大于本身序列的LIS，那它一定就是答案。

Codeforces 605D

但是也有可能 LIS 中不包含 x。

比如对于 A = [1, 2, 3, 100, 4, 5]，LIS 是 5.

如果我们把 A[4] 修改成 99，包含 A[4] 的 LIS 是 1.

但显然此时序列的 LIS 仍然是 5.

如果我们把 A[1] 修改为 200，包含 A[1] 的 LIS 是 1.

但此时序列的 LIS 变成了 4.

Codeforces 605D

通过这两个例子，我们可以发现：

修改的数如果必须在原数列的LIS中：

不包含修改的数的 LIS 会比原来少 1.

修改的数如果可以不在原数列的LIS中：

不包含修改的数的 LIS 不会发生变化.

Codeforces 605D

问题变成了，对于A中的每个数，判断是否必须出现在 LIS 中。

用之前的思路，计算包含 A[i] 的 LIS：

$$\begin{aligned} lis[i] = & \max\{ left[k] \mid k < i,\ A[k] < A[i] \} \\ & + 1 \\ & + \max\{ right[k] \mid k > i,\ A[k] > A[i] \} \end{aligned}$$

如果 lis[i] 等于 A 的 LIS，那么 i 可能出现在 LIS 中。

Codeforces 605D

如果 i 可能出现在 LIS 中：

如果存在 j < i 且 A[j] >= A[i] 且 j 可能出现在 LIS 中，那么 i 可以不出现在 LIS 中。

如果存在 j > i 且 A[j] <= A[i] 且 j 可能出现在 LIS 中，那么 i 可以不出现在 LIS 中。

如果 i 不满足 可以不出现在 LIS 中的条件，那么 i 一定要出现在 LIS 中。

Codeforces 605D

对于每次修改 A[i]=x，先计算 max{ left[k] | k < i, A[k] < x } + 1 + max{ right[k] | k > i, A[k] > x }

如果比 A 的 LIS 大，他就一定是答案。

否则，如果 i 可以不出现在 LIS 中，LIS 不变；

如果 i 一定要出现在 LIS 中， LIS会减一。

Codeforces 605D

遗留问题：计算 max{ left[k] | k < i, A[k] < x }。

经典的二维数点问题。

对 A[k], x 这些值排序。

从小到大枚举 x，把 A[k] < x 的 left[k] 插入树状数组的第 k 个位置。

这里求的是前缀最大值。

只要在更新和询问树状数组的时候，把加法改成取 max 即可。

Codeforces 605D

预处理 left, right 数组。

计算 lis 数组。

离线读入所有询问，对所有的操作 A[i]=x 算一遍包含 x 的LIS。

后两步需要用到树状数组。

时间复杂度 O((n + m) log n)

下节课再见