一类算法复合的方法

江苏省扬州中学 张煜承

问题描述

- 维护集合 S，初始时为空。有 N 个操作需要依次处理
- B X 在 S 中插入一个整数 X
- A Y 询问 S 中被 Y 除余数最小的数，如果有多个则任取一个
- $1 \le N \le 40000$， $1 \le X, Y \le R = 500000$
- 允许离线算法

初步分析

- 算法 1：对询问中每个不同的 Y，维护它对应的询问当前的答案
- 时间复杂度为 $O(N^2)$，不能解决问题
- 但当询问中出现的不同 Y 的个数比较少时会很快，时间复杂度可以写成 $O($不同 Y 的个数 $\times N)$

进一步分析

- 当遇到一个询问 "A Y" 时，要在 S 中寻找使得 $x \bmod Y$ 最小的数 x
- 把这里的 x 写成 $kY+r$，其中 $0 \le r<Y$， k 和 r 是整数
- 也就是说，我们要在集合 S 中，寻找使得 r 最小的数 $kY+r$
- 算法 2：枚举 k，找 $[kY,(k+1)Y)$ 中的最小值。最后在这些最小值中取最优的

一个例子

- S={2,3,6,8} Y=5

0	1	2	3	4	5	6	7	8	9	10	…

最小值为 2　　最小值为 6

- 2 mod 5 = 2
  6 mod 5 = 1
- 因此取 6

- 现在的问题：询问 S 中给定区间 $[a,b]$ 内的最小数
- 可以看成是询问 $\geq a$ 的最小数 $q(a)$

a	0	1	2	3	4	5	6	7	8	9	10	…
$q(a)$	2	2	2	3	6	6	6	8	8	$+\infty$	$+\infty$	…

- 对很多连续的 a， $q(a)$ 是相等的
- 形成了若干个区间

- 假设 X 所在的区间为 $[s,t]$，现在在 S 中插入 X

a	s	$s+1$	…	$X-1$	X	$X+1$	…	$t-1$	t
$q(a)$	X	X	X	X	X	t	t	t	t

- $[s,t]$ 被拆分成了区间 $[s,X]$ 和 $[X+1,t]$

- 只有插入操作，所以一直在拆分区间，而不合并区间
- 让时间倒流，把所有操作按照从后往前的顺序处理，那么区间就一直都在被合并了
- 并查集
- 把这里每个区间看作是一个集合，并维护它们对应的 q
- 每次操作近似地认为是均摊 O(1)

算法 2

- 对一个询问“A Y”，需要询问 O(*R*/*Y*) 个区间，最多 O(*R*) 个区间
- 一次询问的时间复杂度高达 O(*R*)
- 总时间复杂度 O(*NR*)，也不能解决问题

尝试着优化

- 算法 2 的瓶颈在于一次询问需要处理的区间可能非常多，但只会发生在很少当 Y非常小的时候
- 一个例子：当 $Y>R^{0.5}$时，算法 2 已经可以接受了
- 我们可以对这部分很少的 Y的询问使用另一种算法
- 算法 1 当询问中的不同的 Y很少时会很快，所以这里的另一种算法可以选择算法 1

算法 3

- 设一个边界值 K
- 对 $Y>K$ 的询问使用算法 2 $O(NR/K)$
  对 $Y\le K$ 的询问使用算法 1 $O(NK)$
- 总时间复杂度 $O(NK+NR/K)$
- 将 N 和 R 看作常数，容易得出当 $K=R^{0.5}$ 时总时间复杂度最小，为 $O(NR^{0.5})$
- 算法 3 可以完全解决本题

- 我们解决本题的重点是：不使用统一的算法，而是同时使用这个问题的两种算法，分别解决问题中的两个互补的部分

- 我的论文里还有另一个例子
SPOJ RECTANGL
这个问题同样可以通过同时使用两种不同的算法得到较好的解决

总结

- 一个问题往往可以被看作是由若干个相对并列的部分组成起来的
- 通常对这些部分使用统一的算法
- 而有时这个问题可以使用多种算法解决，并且当这些算法应用在问题中不同特征的部分时，会有不同的效果
- 这时就可以将这些算法复合，对问题的不同部分，根据它们的特征分别选择使用对这个部分较优的算法

总结

- 两个算法合并起来后，很神奇地得到了一个更优的算法
- 这是因为这两种算法具有互补的优势，而我们把问题分成了若干部分，对每一部分根据其特征使用较优的算法，就使得两种算法的优势得到了结合

总结

- 每个算法都有各自的优势和劣势
- 如果我们取长补短，充分利用它们的优势，也许就将会得出总体更优的算法
- 这种取长补短的思想是非常重要的