矩形计算解题报告

【简要描述】

给出一个数字矩阵，每次询问一个子矩形中，所有数字出现次数的平方和是多少。

【分析与算法设计】

注：由于 n 和 m 同阶，以下复杂度分析中的 m 均由 n 代替。

算法一：

手算，题目给出了 1 个输入数据，不难算出这个数据点的答案。期望得分 5 分。

时间复杂度：O(1)

算法二：

暴力，对于每一个询问，我们枚举每一个在原矩阵出现的数字，统计这个数字在询问的子矩阵中出现多少次，并且把出现次数的平方统计进答案。期望得分 5 分。

时间复杂度：$O(qn^4)$

算法三：

考虑对算法二进行改进。我们没有必要枚举每一个数字。对于某个数字的出现次数我们可以用 hash 表来统计，对于答案的更新我们也可以实时维护，即若一个数字由出现 x 次变成出现 x+1 次，答案增加 2x+1。经过这样的优化以后，程序效率大幅提高，期望得分 30 分。

时间复杂度：$O(qn^2)$

算法四：

在线地回答询问，似乎碰到了瓶颈，为了获得更大的优化空间，我们来考虑离线回答询问。对于同一个数字，它对于每一个询问的答案的贡献，我们一起算。我们先枚举当前统计哪个数字，然后把是这个数字的地方标记成 1，不是这个数字的地方标记成 0，每次询问相当于询问一个子矩阵中 1 的个数。这个问题我们可以通过类似在一维的时候求前缀和的方法求从(1,1)-(x,y)的子矩阵和。然后每次询问就可以在 O(1)的时间内回答了，期望得分 60 分。

时间复杂度：O(不同数字个数*(n^2+q))

算法五：

分析算法四，它之所以可能会超时的原因是当数字种数特别多的时候就退化成 $O(n^4+qn^2)$了。那么我们是否可以让一部分数字采用算法四的方法，另一部分用别的方法来做呢？我们定一个界 p，并且作如下规定：情况 1，如果一个数字出现次数>=p，我们执行

算法四算该数字对每个询问的答案的贡献。情况 2，如果一个数字出现次数$<p$ 我们采用我接下来将会描述的算法。由于满足情况 1 的数字种数不会超过 n^2/p，所以这里的复杂度最多是 $n^2/p*(n^2+q)$。我们现在来讨论情况 2。由于每种数字出现次数都很少，所以数字种类可能很多，我们想对每种数字分开来算贡献可能非常困难，继续往这个方面想下去也许很难有突破口。我们需要换一种思路。注意题目要求算的是每种数字出现次数的平方求和，我们似乎可以从这里下手。若有一个询问，对于某种数字，有 3 个位置 A,B,C 处于这个询问的矩形内部，既然我们无法统计出现次数，那么我们直接统计出现次数的平方。为了统计出现次数的平方我们可以统计有多少个位置对处于询问矩阵中。上面那个例子，我们不直接统计 A,B,C。改成统计(A,A),(A,B),(A,C),(B,A),(B,B),(B,C),(C,A),(C,B),(C,C)。于是我们的算法出来了，对于一种出现次数$<p$ 的数字，我们枚举出现位置对(A,B)，然后对所有包含这一对位置的询问矩阵的询问的答案+1，由于属于这种情况的数字，它们出现的次数都$<p$ 所以最多有 $p^2n^2/p=pn^2$ 个位置对。至此这个问题似乎已经解决了，但是细心的读者可以发现我们到底如何来快速的让所有包含这一对位置的询问的答案+1？对于一个位置对(A,B)它有四个关键字(x1,y1,x2,y2)其中 x1=min(Ax,Bx),y1=min(Ay,By),x2=max(Ax,Bx),y2=max(Ay,By)。如果一个矩阵(a,b)-(c,d)要包含这个位置对那么必须要满足四个不等式 $a<=x1,b<=y1,c>=x2,d>=y2$，对于这样的带约束条件的统计问题，我们很容易想到用四维树状数组来解决。然而这样空间复杂度是承受不了的，有 n^4 的级别，时间复杂度更是有 $\log^4 n$，为了解决空间问题也为了减少时间复杂度，我们可以先将一维关键字进行排序，也就是说我们可以先按 x1 排序，先保证了 $a<=x1$ 这个条件以后，剩下三维再用三维树状数组解决。于是空间复杂度成功降到 n^3，时间复杂度降到 $\log^3 n$。至此情况 2 的算法就完成了。我们来分析一下情况 2 的最坏时间复杂度，上面已经分析了位置对最多有 pn^2 个，对于每个位置对的插入是 $\log^3 n$ 的，对于每个询问的查询也是 $\log^3 n$ 的，那么这里的总时间复杂度就是$(pn^2+q)\log^3 n$。结合情况 1 和情况 2，我们希望两种情况的复杂度尽量均匀也就是 $pn^2\log^3 n=n^2/p*(n^2+q)$，从这个方程我们可以解出合适的 $p=\sqrt{(n^2+q)/\log^3 n}$。如果在分块时我们忽略 q 的话，时间复杂度就是 $O(n^3\log^{1.5}n+q(n\log^{1.5}n+\log^3 n))$。虽然算出来极限数据的复杂度还是有几亿，但是如果不是根据所选用的 p 值刻意去构造，是很难达到上限的，对于所有数据都能很快出解，至此本题完美解决。

时间复杂度：$O(n^3\log^{1.5}n+q(n\log^{1.5}n+\log^3 n))$