方格取数

东北师大附中 王康宁

摘 要

2013年集训队互测论文，一道趣题的解题报告。

1 题目

1.1 题目描述

有一个$N*N$的方格表，每个格子里有一个数，第i行第j列的数等于A_i*B_j。一个人从左上角走到右下角，每次只能向右或向下走一步，他会把经过的数累加起来，并使这个总和最小。他决定走T次，第i次从第P_i 行第Q_i列的点走到第R_i行第S_i列的点，问每一次经过的数的最小总和。

1.2 输入格式

第一行包括两个整数N,T，分别表示方格表的大小和询问数量。
第二行N个整数，表示$A_1,A_2,...,A_N$。这一行的数是随机生成的。
第三行N个整数，表示$B_1,B_2,...,B_N$。这一行的数是随机生成的。
接下来T行，每行四个整数$P_i,Q_i,R_i,S_i(P_i \le R_i,Q_i \le S_i)$，含义如题所述。

1.3 输出格式

输出T行，每行一个整数，表示答案。

1.4 输入样例

3 4

1 2 3
2 3 4
1 1 1 1
1 3 1 3
1 1 3 3
1 1 3 1

1.5 输出样例

2
4
29
12

1.6 数据范围

对于20%的数据，$N \leq 100, T \leq 100$。
对于50%的数据，$N \leq 3000, T \leq 1000$。
对于70%的数据，$N \leq 50000, T \leq 20000$。
对于100%的数据，$N \leq 200000, T \leq 50000, 1 \leq A_i, B_i \leq 10^6$。保证询问合法，除样例外，所有的Ai, Bi均为随机生成的。

1.7 时间限制

2秒。

1.8 空间限制

512MB。

2 解题思路

2.1 算法一

对于每一组询问，用动态规划直接求解。

时间复杂度：$O(T*N*N)$
空间复杂度：$O(N*N)$
期望得分：20分

2.2 算法二

上一个算法的询问太慢了，现在需要加速询问。

考虑利用分治思想。每次按照中线将矩形分成两半，用动态规划求出中线上每个点到整个矩形上每个点的答案，并递归处理中线两侧的矩形。如果所询问的起点和终点在中线的两侧，则枚举经过的中线上的点，否则递归到某个子矩形。

如果每次都沿竖线切割的话，设预处理时间复杂度为$Time(N)$。则

$$Time(x) = 2 * Time(x/2) + O(N * N * x) \tag{1}$$

这样预处理时间复杂度为$O(N^3 * logN)$。而如果每次交替沿横线和竖线切割，则

$$Time(x) = 2 * Time(x/2) + O(x^3) \tag{2}$$

预处理时间复杂度为$O(N^3)$。

总时间复杂度：$O(N^3 + T * N)$
空间复杂度：$O(N^3)$
预处理复杂度过高，只能尝试其他方案。

2.3 算法三

考虑分块算法，如果将这个矩形分成S*S块，预处理每个点到块边界的每个点的答案，以及每个块内每对点的答案。

询问时两个点若不在同一块，则枚举其中一个点第一次经过块边界的位置，否则直接回答。

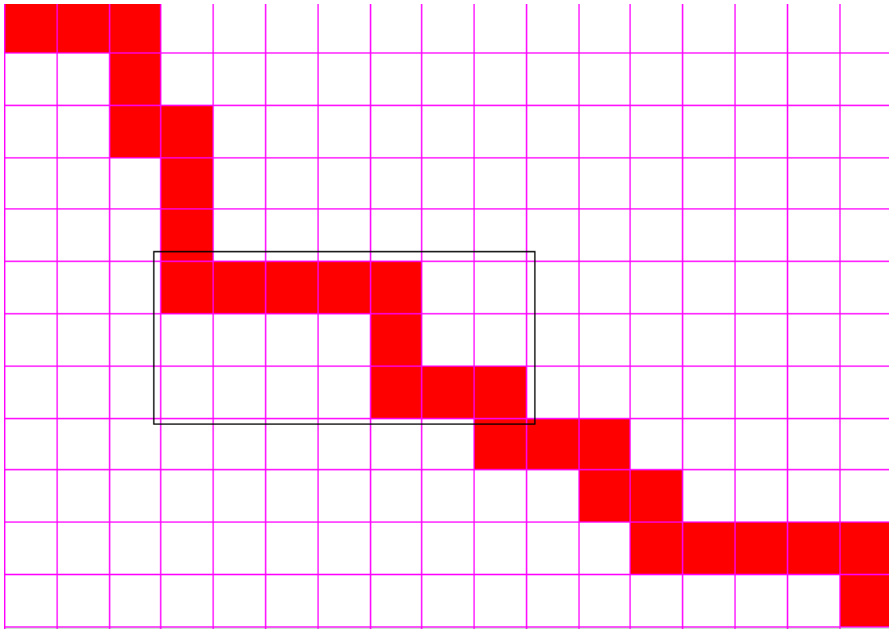
时间复杂度：$O(N^3 * (S + N/S^2) + T * N/S)$
空间复杂度：$O(N^3 * (S + N/S^2))$
复杂度同样过高。

2.4 改进方案

上述算法低效的一个主要原因是没有注意到题目中的一个重要条件：序列A和B均为随机生成的。

经过尝试和观察，可以发现答案的转折点不会很多。

注意一个转折点，和它相邻的两个转折点之间的部分。如图：

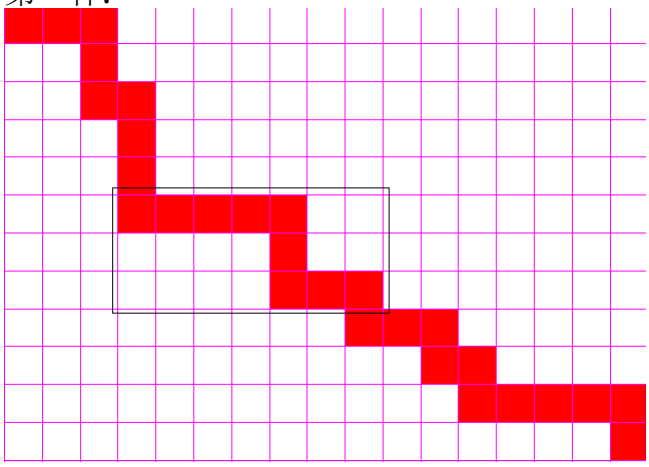


假如这个转折点所在列的权值不小于区域中其左边的某列权值，也不小于区域中其右边的某列权值。下面给出一种不更差的方案。

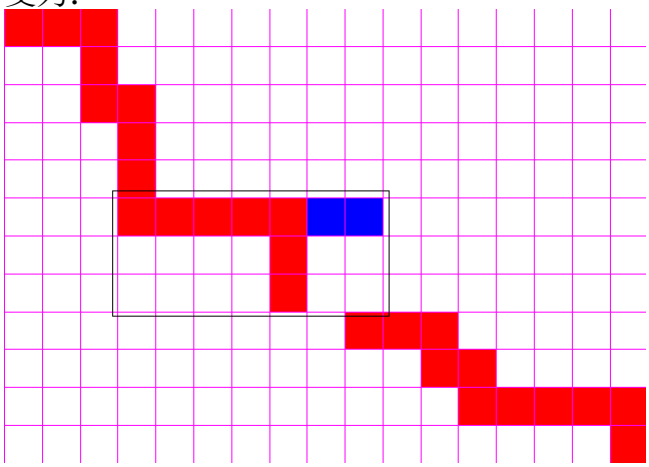
假设图中方框内第五列权值不小于方框最左和最右列的权值。

考虑下列两种变换方案：

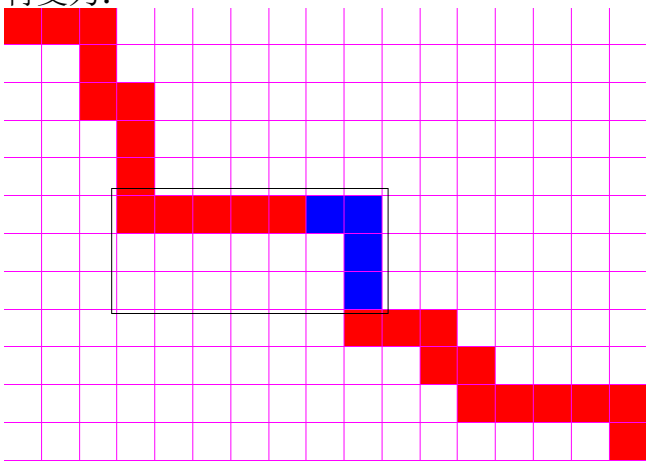
第一种：



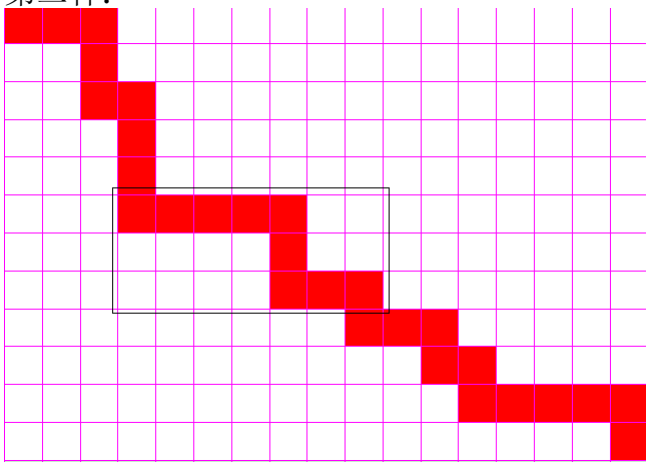
变为：



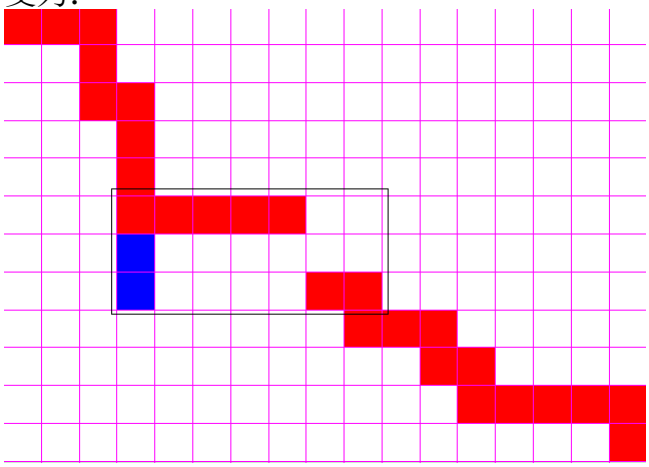
再变为：



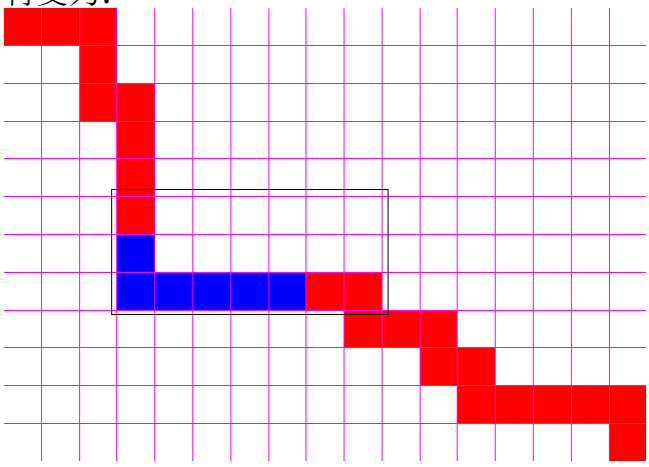
第二种：



变为：



再变为：



如果区域中上面一行权值小于下面一行的话，则经过第一种变换方案答案更优。

如果区域中上面一行权值大于下面一行的话，则经过第二种变换方案答案更优。

如果区域中上面一行权值等于下面一行的话，则经过任何一种变换方案之后答案不会变差。

这样我们就可以消除这个转折点了。

所以，设转折点分别在$L_1, L_2, ..., L_k$行（列），一定存在一种最优方案，使得：L_i行的权值，要么小于L_{i-1}至$L_i - 1$所有行的权值，要么小于$L_i + 1$至L_{i+1}所有行的权值。

若前者成立，则L_{i-1}行的权值一定小于L_{i-2}至$L_{i-1} - 1$所有行的权值，而L_{i-2}行的权值一定小于L_{i-3}至$L_{i-2} - 1$所有行的权值，...，所以L_i行权值比之前所有行的权值都小。

因此，任意一个转折点所在行（列）的权值，要比询问区域中左边所有行（列）权值都小，或者比右边所有行（列）权值都小。

而对于随机数据，第i行的权值比之前所有行权值都小的概率为$1/i$。

因此期望的转折点个数为：$1/1 + 1/2 + ... + 1/N = O(logN)$。

2.5 算法四

我们暴力找出所有可能存在转折点的行和列。

然后动态规划求解，此时，只有可能存在转折点的行和列才为有效状态，期望行数和列数都是$O(logN)$，而期望的乘积等于乘积的期望，所以期望状态数是$O(log^2N)$。

记录数组A和B的部分和来加速转移。

时间复杂度（期望）：$O(T * (N + log^2N)) = O(T * N)$

期望得分：50分

2.6 算法五

上个算法的瓶颈是找出可能存在转折点的行（列），即询问一个序列某区间比之前（之后）所有数都小的数，只要求出每个数之后（之前）第一个比它小的数即可，而这一工作可以使用单调队列来完成。

比如求每个数之后第一个比它小的数，可以从后向前扫描这个序列，将单调队列队尾不小于当前数的部分出队，则这个数之后第一个比它小的数就是当前队尾的数，然后将当前数入队。

预处理只要线性时间，每次询问时间为$O(log^2N)$。

时间复杂度（期望）：$O(N+T*log^2N)$

期望得分：100分

至此，我们已完满解决了这个问题。

3 致谢

感谢CCF提供学习和交流的平台。

感谢孔维玲老师和谷方明老师对我的指导。

感谢帮助过我的同学们。