《Matlab与物流管理实验》作业: Monge矩阵及应用

This is the final project. Submit this report before 6pm on May 28, 2019.

如果一个$m \times n$矩阵A的元素满足以下不等式，我们就称它为Monge 矩阵（或Monge 数组）：

$$A(i,j) + A(k,\ell) \le A(i,\ell) + A(k,j), \forall 1 \le i < k \le m, 1 \le j < \ell \le n. \tag{1}$$

直观上，Monge矩阵的任意两行(即第i行和第k行)两列(即第j列和第ℓ列)组成的2×2子矩阵，其对角线元素和，不超过其副对角性元素和。这是应用数学和计算机学科中一类重要的矩阵，最早被法国数学家Gaspard Monge研究，它实质上反应了所谓的次模性(submodular)数据结构。关于此类矩阵及其性质的更多介绍，请自行搜索维基百科。

以下是维基百科相应页面给出一个Monge矩阵的例子：$\begin{bmatrix} 10 & 17 & 13 & 28 & 23 \\ 17 & 22 & 16 & 29 & 23 \\ 24 & 28 & 22 & 34 & 24 \\ 11 & 13 & 6 & 17 & 7 \\ 45 & 44 & 32 & 37 & 23 \\ 36 & 33 & 19 & 21 & 6 \\ 75 & 66 & 51 & 53 & 34 \end{bmatrix}$。如选取第$2,4$行以及第$1,5$列构成的子矩阵其对角线元素和$17+7$显然不超过副对角线元素和$11+23$。

任务一：理论分析

Monge矩阵的原始定义要求不等式(1)对任意两行两列都成立，因此检验一个$m \times n$的矩阵是否为Monge矩阵总共需要$O(m^2n^2)$次比较。Monge矩阵的的另一种定义是满足如下不等式的矩阵：

$$A(i,j) + A(i+1,j+1) \le A(i,j+1) + A(i+1,j), \forall 1 \le i < m, 1 \le j < n. \tag{2}$$

这一定义只需要检验$O(mn)$次。请问，如果A的元素满足(2)式，那么这些元素也满足(1)式吗？请从理论上证明这一点。提示：先考虑一个类似的问题，序列$\{x_n : n \ge 1\}$对n递增的定义既可以是“$x_i \le x_j$对任意$1 \le i < j$成立”，也可以是“$x_i \le x_{i+1}$对任意$i \ge 1$成立”。另外，如果A是Monge矩阵，请问它的转置A'是否仍是Monge矩阵？请判断并简要说明理由。

任务二：Monge矩阵的验证和生成

在Matlab中实现函数isMonge(A) 用于判断矩阵A是否是Monge矩阵，以及函数A=genMonge(m,n)用于随机生成元素为自然数的$m \times n$的Monge矩阵A。要求：尽量避免使用循环命令判断Monge矩阵，可以使用diff函数；另外，请在报告中说明生成随机Monge矩阵的思路。

任务三：Monge矩阵与优化

在很多应用问题中，我们可能需要寻求矩阵中的最小元素位置。具体而言，对于任一$m \times n$矩阵A, 需要寻找位置(a,b)使得$A(a,b) = \underset{1\le i\le m, 1\le j\le n}{\text{minimize}} A(i,j)$. 为此，Matlab已经内置了函数min。请查阅此命令的手册，编写函数[a,b]=getMin0(A), 实现上述需求。

由于要遍历矩阵的所有元素，此算法的效率一般为$O(mn)$. 然而，如果目标矩阵A是Monge矩阵，则有可能更高效地找到其最小元素位置，其关键在于Monge矩阵的重要性质：行最小值所在列数（对行数来说）递增。具体而言，如果Monge矩阵第i行最小元素在第$j(i)$列，那么$j(i)$对i递增。比如，对前述维基百科给出例子，从第1行开始，其各行最小值所在列数分别为$1,3,3,3,5,5,5$，这显然是一个递增序列。因此，基于此性质，我们只需要先寻求Monge矩阵第1行最小值列数$j(1)$；在寻找第2行最小值列数时，可忽略第$1,2,\cdots,j(1)-1$)列元素，直接从第$j(1)$列开始往后查找$j(2)$；然后再按同样的方式依次寻找第$3,4,\cdots$行的最小值列数。

上述过程中，如果足够幸运，处理第1行时得到的$j(1)$可能就是此矩阵的最后一列，即$j(1)=n$. 由于$j(1)\le j(2)\le\cdots\le j(m)$ 而且$j(i)\le n$对所有行i都成立，因此$j(i)\equiv m$, 即所有行的最小值一定都位于最后一列，因此整个过程只需要遍历$O(m+n)$个元素。当然，也可能不够幸运，所有行的最小值刚好都位于第1列，导致在检验每一行时，都要遍历所有n列后，才能够确定最小值其实一直在第1列。这一过程可能仍然需要$O(mn)$次检验。

尽管如此，我们可以通过技术手段避免一些“不够幸运”的情况。注意到Monge矩阵的转置仍然是Monge矩阵（见任务一），各列最小值所在行数（对列数来说）递增。因此可以先耗费$O(m+n)$次比较，寻找到第1行最小值的列坐标$j(1)$和第1列最小值的行坐标$i(1)$，然后下一轮搜索时只针对此矩阵第$(i(1),j(1))$右侧和下文元素构成的子矩阵，寻找其最小元素。

请以上述思路为基础，编写针对Monge矩阵的函数[a,b]=getMin1(A)得到其最小元素的位置。然后通过数值形式，与前面得到的getMin0函数比较两者的效率。要求：自行设计实验方案，清楚地描述思路和目标，然后在Matlab中通过足够的数值例子进行验证。

最后，作为可选任务，思考如下问题：万一第i行和第i列的最小值刚好都在(i,i)位置，上述过程是否仍可能需要$O(mn)$次检验？有什么办法可进一步提升效率？另外注意，之前关于DELS问题的作业中，提到了利用$\ell(i,j)=K_i+a_ix_j$的边长结构，可以提升寻找最短路径算法的效率；其关键在于这些边长构成的矩阵，刚好是一个Monge矩阵。