

# 高等理工学院《算法设计与分析》

## (2019 年秋季学期)

### 第三次作业

作业提交截止时间：2019 年 11 月 26 日 23 : 55

#### 1 行列均衡问题 (20 分)

给定一个  $n$  阶方阵  $A$  ( $n$  行  $n$  列的矩阵)，可以对其执行如下操作：

- 任选该方阵中的一个元素并将其值加 1。

请你设计算法计算使该方阵的每一行之和与每一列之和均相等所需的最少操作次数。

例如，方阵  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  可以通过四次操作变为  $\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}$ ，此时该方阵各行各列之和均为 7。

#### 2 数据修改问题 (20 分)

给定一个长度为  $n$  的数组  $a[1..n]$ 。最初， $a$  的所有元素都等于 0。

现要对  $a$  中的数据进行  $Q$  次修改，每次修改给定三个数  $l_i, r_i$  和  $x_i$  ( $1 \leq i \leq Q$ )，其中  $1 \leq l_i \leq r_i \leq n$  且  $x_i > 0$ ，表示将数组  $a[l_i..r_i]$  中的所有元素均加上  $x_i$ 。

现允许你从这  $Q$  次修改中删掉一次，使得删掉这次修改后，执行剩余的修改得到的数组  $a$  中的最大元素尽可能小。请求出该最小值。

例如，对于一长度为 4 的数组  $[0, 0, 0, 0]$ ，有三次修改分别为  $\begin{cases} l_1 = 2, r_1 = 4, x_1 = 2 \\ l_2 = 1, r_2 = 2, x_2 = 2 \\ l_3 = 3, r_3 = 4, x_3 = 1 \end{cases}$ 。经过

第一次修改后，该数组变为  $[0, 2, 2, 2]$ ；经过第二次修改后，该数组变为  $[2, 4, 2, 2]$ ；经过第三次修改后，该数组变为  $[2, 4, 3, 3]$ 。

如果我们删掉第 1 次修改，则经过 2、3 次修改得到的数组为  $[2, 2, 1, 1]$ ，该数组中的最大值为 2。可以验证这是在所有可行方案中数组的最大值最小的方案。（删掉第二次修改得到的数组为  $[0, 2, 3, 3]$ ，最大值为 3；删掉第三次修改得到的数组为  $[2, 4, 2, 2]$ ，最大值为 4。）

#### 3 最大矩阵问题 (20 分)

给定一个  $n$  行  $m$  列的二进制矩阵  $M$  (对于  $1 \leq i \leq n, 1 \leq j \leq m, M_{i,j} = 0$  或  $1$ )。可以对其执行如下两种的操作：

1. 选择矩阵的第  $i$  行，并将该行的每个元素取反（如果该元素为 0 则将其改为 1，如果该元素为 1 则将其改为 0）；
2. 选择矩阵的第  $j$  列，并将该列的每个元素取反。

例如，矩阵  $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$  将第一行取反可以得到  $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ ；在此基础上再将第三列取反可以得到  $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ 。每种操作都可以执行无限次。

最终，将得到的矩阵的每一行解释为一个数字的二进制表示，例如，矩阵  $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$  的第一行为二进制数  $(101)_2$ ，其对应的十进制数为  $4+0+1=5$ ，第二行为二进制数  $(111)_2$ ，其对应的十进制数为 7，第三行为二进制数  $(110)_2$ ，其对应的十进制数为 6，这三个数之和为  $5+7+6=18$ 。现请你计算在通过执行上述操作可以得到的矩阵中， $n$  个二进制数之和的最大值。

## 4 环路问题 (20 分)

给出一个联通无向图  $G = (V, E)$ ，请设计一种尽可能高效的算法来判断  $G$  中是否有环。如果有，请输出任意一个环 (按顺序给出环上的每个顶点)。请解释算法的正确性并分析算法的时间复杂度。

## 5 最小生成树问题 (20 分)

给定一个无向连通图  $G = (V, E)$ ，其中每条边的权值只可为 1 或 2。请设计一个时间复杂度为  $O(|V| + |E|)$  的算法来求  $G$  的一棵最小生成树，并验证其时间复杂度。