

# 算法设计与分析

CS2008 班 U202015533 徐瑞达

2022.04.20

## 目录

<b>1</b>	<b>22-25 章作业</b>	<b>2</b>
1.1	计算题 . . . . .	2
1.1.1	24.1-1 . . . . .	2
1.1.2	24.4-1 . . . . .	3
1.1.3	25.2-1 . . . . .	3
1.2	设计、证明题 . . . . .	5
1.2.1	24.1-3 . . . . .	5
1.2.2	24-3 . . . . .	5
1.2.3	25.2-7 . . . . .	6
1.3	思考题 . . . . .	6
1.3.1	25.2-6 . . . . .	6

# 1 22-25 章作业

## 1.1 计算题

### 1.1.1 24.1-1

Question:

在图 24-4 上运行 *Bellman – Ford* 算法，使用结点  $z$  作为源结点。在每一遍松弛过程中，以图中相同的次序对每条边进行松弛，给出每遍松弛操作后的  $d$  值和  $\pi$  值。然后，把边  $(z, x)$  的权重改为 4，再次运行该算法，这次使用  $s$  作为源结点。

Answer:

使用  $z$  作为源结点，则得到每遍松弛操作后的值如下：

表 1:  $d$  的值

$s$	$t$	$x$	$y$	$z$
$\infty$	$\infty$	$\infty$	$\infty$	0
2	$\infty$	7	$\infty$	0
2	5	7	9	0
2	5	6	9	0
2	4	6	9	0

表 2:  $\pi$  的值

$s$	$t$	$x$	$y$	$z$
NIL	NIL	NIL	NIL	NIL
$z$	NIL	$z$	NIL	NIL
$z$	$x$	$z$	$s$	NIL
$z$	$x$	$y$	$s$	NIL
$z$	$x$	$y$	$s$	NIL

改变边  $(z, x)$  的权重为 4 后，使用  $s$  作为源结点，则得到每遍松弛操作后的值如下：

表 3:  $d$  的值

$s$	$t$	$x$	$y$	$z$
0	$\infty$	$\infty$	$\infty$	$\infty$
0	6	$\infty$	7	$\infty$
0	6	4	7	2
0	2	4	7	2
0	2	4	7	-2

表 4: $\pi$ 的值				
s	t	x	y	z
NIL	NIL	NIL	NIL	NIL
NIL	s	NIL	s	NIL
NIL	s	y	s	t
NIL	x	y	s	t
NIL	x	y	s	t

### 1.1.2 24.4-1

**Question:**

请给出下面差分约束系统的可行解或证明该系统没有可行解。

$$\begin{aligned}
 x_1 - x_2 &\leq 1 \\
 x_1 - x_4 &\leq -4 \\
 x_2 - x_3 &\leq 2 \\
 x_2 - x_5 &\leq 7 \\
 x_2 - x_6 &\leq 5 \\
 x_3 - x_6 &\leq 10 \\
 x_4 - x_2 &\leq 2 \\
 x_5 - x_1 &\leq -1 \\
 x_5 - x_4 &\leq 3 \\
 x_6 - x_3 &\leq -8
 \end{aligned}$$

**Answer:**

设约束图的顶点是

$$v_0, v_1, v_2, v_3, v_4, v_5, v_6$$

边为

$$(v_0, v_1), (v_0, v_2), (v_0, v_3), (v_0, v_4), (v_0, v_5), (v_0, v_6), (v_2, v_1), (v_4, v_1), (v_3, v_2), (v_5, v_2), (v_6, v_2), (v_6, v_3)$$

对应权重分别为

$$0, 0, 0, 0, 0, 1, -4, 2, 7, 5, 10, 2, -1, 3, -8$$

计算

$$(\delta(v_0, v_1), \delta(v_0, v_2), \delta(v_0, v_3), \delta(v_0, v_4), \delta(v_0, v_5), \delta(v_0, v_6))$$

得到

$$(-5, -3, 0, -1, -6, -8)$$

而该方案是可行的。

### 1.1.3 25.2-1

**Question:**

在图 25-2 所示的带权重的有向图上运行 *Floyd – Warshall* 算法，给出外层循环的每一次迭代所

生成的矩阵  $D^{(k)}$ 。

**Answer:**

$k = 1 :$

$$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$$

$k = 2 :$

$$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$$

$k = 3 :$

$$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$$

$k = 4 :$

$$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

$k = 5 :$

$$\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

$k = 6 :$

$$\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

## 1.2 设计、证明题

### 1.2.1 24.1-3

Question:

给定  $G = (V, E)$  是一带权重且没有权重为负值的环路的有向图, 对于所有结点  $v \in V$ , 从源结点  $s$  到结点  $v$  之间的最短路径中, 包含边的条数的最大值为  $m$ 。(这里, 判断最短路径的根据是权重, 不是边的条数。) 请对算法 *BELLMAN - FORD* 进行简单修改, 可以让其在  $m + 1$  遍松弛操作之后终止, 即使  $m$  不是事先知道的一个数值。

Answer:

由上界理论, 可得经过  $m$  次迭代,  $d$  的值不会被改变。因此, 在第  $m + 1$  次迭代中  $d$  的值不会被迭代。然而, 事先并不知道  $m$  的确切值, 因此无法确切地进行  $m$  次迭代后终止算法。如果尝试当  $d$  的值不会再改变时终止算法, 此时将已经进行了  $m + 1$  次迭代。

### 1.2.2 24-3

Question:

套利交易指的是使用货币汇率之间的差异来将一个单位的货币转换为多于一个单位的同种货币的行为。例如, 假定 1 美元可以购买 49 印度卢比, 1 印度卢比可以购买 2 日元, 1 日元可以购买 0.0107 美元。那么通过在货币之间进行转换, 一个交易商可以从 1 美元开始, 购买  $49 \times 2 \times 0.0107 = 1.0486$  美元, 从而获得 4.86% 的利润。假设给定  $n$  种货币  $c_1, c_2, \dots, c_n$  和一个  $n \times n$  的汇率表  $R$ , 一个单位的  $c_i$  货币可以购买  $R[i, j]$  单位的  $c_j$  货币。  
a. 给出一个有效的算法来判断是否存在一个货币序列  $\langle c_{i_1}, c_{i_2}, \dots, c_{i_k} \rangle$ , 使得  $R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1$  请分析算法的运行时间。  
b. 给出一个有效算法来打印出这样的一个序列 (如果存在这样一种序列)。分析算法的运行时间。

Answer:

a. 通过考虑以下步骤来实现该算法

$$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_k, i_1] > 1,$$

$$\ln(R[i_1, i_2]) + \ln(R[i_2, i_3]) + \cdots + \ln(R[i_k, i_1]) > 0,$$

$$-\ln(R[i_1, i_2]) - \ln(R[i_2, i_3]) - \cdots - \ln(R[i_k, i_1]) < 0$$

b.

为实现该功能, 首先在 a 的基础上, 对所有边的权重执行相同的修改。然后, 按照 *BellmanFold* 算法, 松弛所有边  $|V| - 1$  次来检测负重循环, 并记录所有顶点的  $d$  值。然后松弛所有边  $|V|$  次并检查哪些顶点的  $d$  值有所减少。所有这些顶点一定位于某些可能不相交的负重循环集中, 称这组顶点集为  $S$ 。为了找到其中一个循环, 可以在  $S$  中选择任何一个顶点, 并贪心地选择与其有边相连

并在  $S$  中的顶点。在这个过程中，永远不会到达死胡同，因为集合  $S$  由处于某些循环中的顶点组成，所以每个顶点的出度至少为 1。

### 1.2.3 25.2-7

Question:

在 *Floyd – Warshall* 算法中构建最短路径的另一种办法是使用  $\phi_{ij}^{(k)}$ ，其中  $i, j, k = 1, 2, \dots, n$ ， $\phi_{ij}^{(k)}$  是从结点  $i$  到结点  $j$  的一条中间所有结点都取自集合  $1, 2, \dots, k$  的最短路径上编号最大的中间结点。请给出  $\phi_{ij}^{(k)}$  的一个递归公式，并修改 *Floyd – Warshall* 过程来计算  $\phi_{ij}^{(k)}$  的值，并重写 *PRINT – ALL – PAIRS – SHORTEST – PATH* 过程，使其以矩阵  $\Phi = (\phi_{ij}^{(k)})$  作为输入。矩阵  $\Phi$  与 15.2 节所讨论的链式矩阵乘法中的表格存在何种相似点？

Answer:

当  $d_{ik}^{(k)} + d_{kj}^{(k)} \geq d_{ij}^{(k-1)}$  时，使  $\phi_{ij}^{(k)} = \phi_{ij}^{(k-1)}$ ，否则  $\phi_{ij}^{(k)} = k$ ，以此递归计算  $\phi_{ij}^{(k)}$ 。修改 *Floyd – Warshall* 过程时，只需使用上述递归公式计算  $\phi_{ij}^{(k)}$  的值，这不会增加其渐进运行时间。已知在  $\phi_{ij}^{(k)}$  中得到的  $n^3$  个值，则可以重新构建从  $i$  到  $j$  的最短路径，因为该路径中最大的顶点为  $\phi_{ij}^{(n)}$ ，记作  $a_1$ 。那么在  $a_1$  之前的路径中的最大顶点为  $\phi_{ia_1}^{(a_1-1)}$ ，在  $a_1$  之后的路径中的最大顶点为  $\phi_{a_1j}^{(a_1-1)}$ ，如此递归直至得到出现的最大元素为 *NIL*，这样可以一直细分路径直至最短路径被完全构建。

## 1.3 思考题

### 1.3.1 25.2-6

Question:

我们怎样才能使用 *Floyd – Warshall* 算法的输出来检测权重为负值的环路？

Answer:

1. 检测结果矩阵主对角线是否存在负值。2. 多执行一遍循环，如果还有矩阵元素更新，表明存在负权重环路