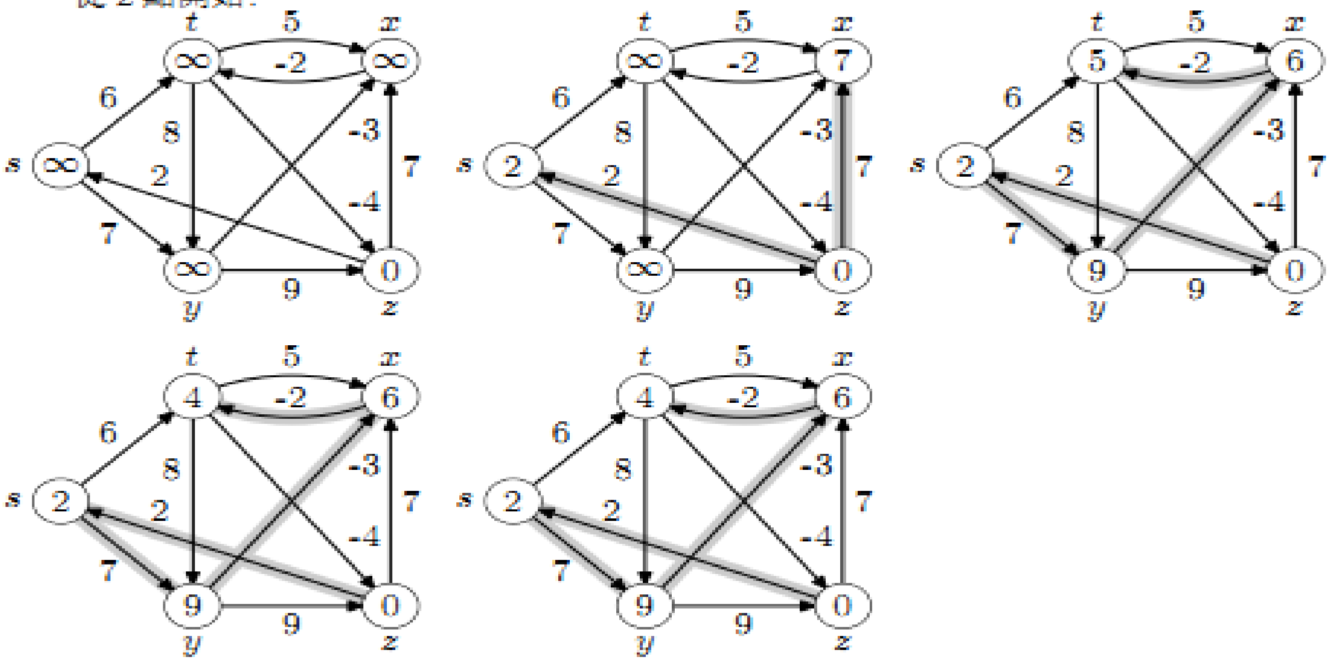
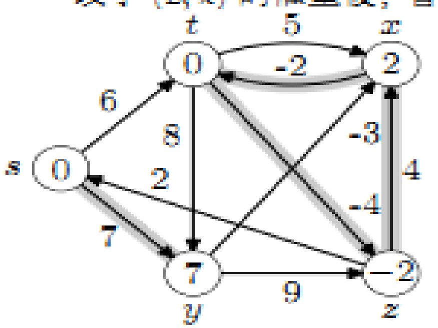
22-25章作业

24.1-1 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 π 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。

從 z 點開始:



改了 (z, x) 的權重後，會出現總權重爲負值的環路。

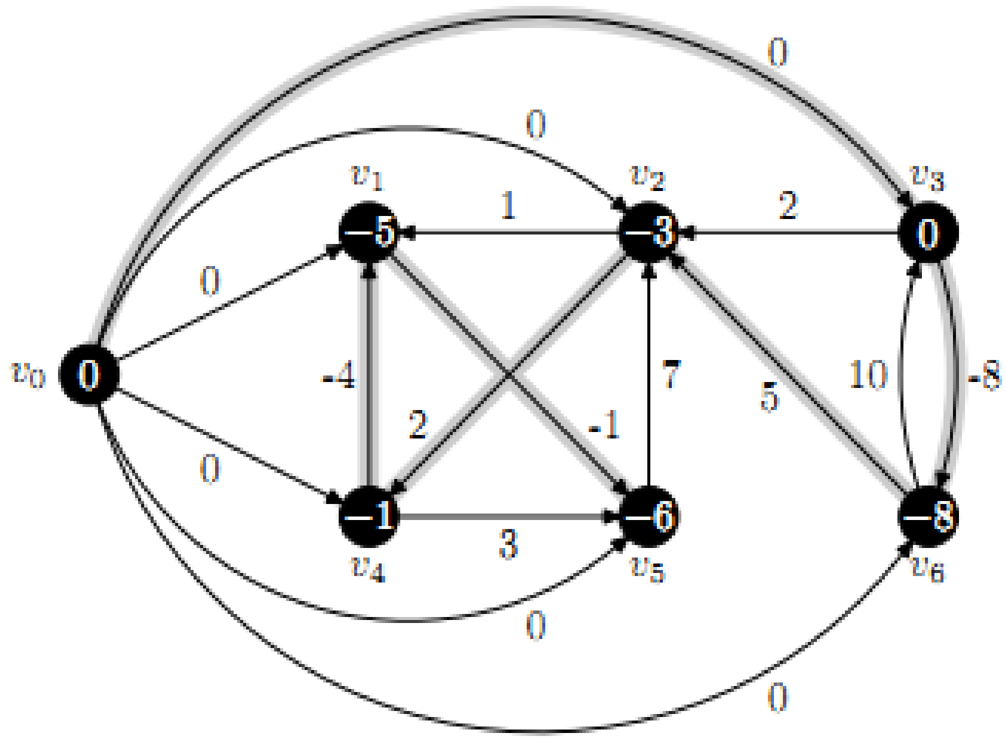


$(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y).$

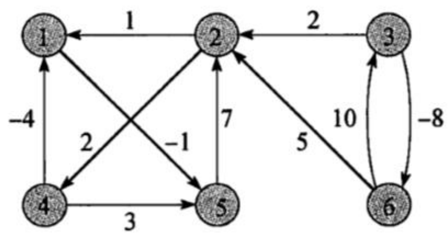
24.4-1 请给出下面差分约束系统的可行解或证明该系统没有可行解。

$$x_1 - x_2 \leqslant 1$$
$$x_1 - x_4 \leqslant -4$$
$$x_2 - x_3 \leqslant 2$$
$$x_2 - x_5 \leqslant 7$$
$$x_2 - x_6 \leqslant 5$$
$$x_3 - x_6 \leqslant 10$$
$$x_4 - x_2 \leqslant 2$$
$$x_5 - x_1 \leqslant -1$$
$$x_5 - x_4 \leqslant 3$$
$$x_6 - x_3 \leqslant -8$$

约束图



25.2-1 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 $D^{(k)}$。



$$D^{(1)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} \quad D^{(2)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix} \quad D^{(4)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix} \quad D^{(6)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

24.1-3 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。

（1）如果权重最小的路径上的边数最多为m，则根据路径松弛性质，在Bellman-ford算法执行了m次迭代后，每个结点v的v.d就会到达了它的最小路径权重。

再根据上界性质，经过m次迭代，就没有哪个结点的d再能被修改了。因此，m+1次循环后，d值不再改变了，算法终止。

（2）如果我们事先不知道m的大小，就不能使算法恰好执行m次然后终止。但是，如果我们可以检查是不是有改变发生，如果没有结点的d值再发生改变，则说明可以使算法终止了。发生最后一次改变的那次循环就是第m次循环。

```
BELLMAN-FORD-(M+1)(G, w, s)
  INITIALIZE-SINGLE-SOURCE(G, s)
  changes = TRUE
  while changes == TRUE
      changes = FALSE
      for each edge (u, v) ∈ G.E
          RELAX-M(u, v, w)

RELAX-M(u, v, w)
  if v.d > u.d + w(u, v)
      v.d = u.d + w(u, v)
      v.π = u
      changes = TRUE
```

注：这里没有检查负权重的环路。如果存在负权重的环路，while循环将无限循环下去。

24-3 （套利交易） **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 $49\times2\times0.0107=1.0486$ 美元，从而获得 4.86%的利润。

假设给定 n 种货币 c_1，c_2，…，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$\langle c_{i_1}, c_{i_2}, \cdots, c_{i_k}\rangle$，使得

$$R[i_1, i_2]\cdot R[i_2, i_3]\cdot\cdots\cdot R[i_{k-1}, i_k]\cdot R[i_k, i_1]>1$$

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。

a. 用bellman-ford算法。

首先建立一个有向图G=<V,E>，每种货币对应一个结点，根据汇率表，若$R[i_1,i_2]$，则从i_1到i_2引一条边。图中增加一个新结点v_0，从v_0引出到其它所有结点的边$<v_0,v_i>$

要使得$R[i_1,i_2]*\cdots R[i_k,i_1]>1$，即有$\lg\frac{1}{R[i_1,i_2]}+\cdots+\lg\frac{1}{R[i_k,i_1]}<0$。

因此，置边(v_i,v_j)权重$w\left(v_i,v_j\right)=lg\frac{1}{R[i,j]}=-lgR[i,j]$。

对v_0，它到其它所有节点v_l的边(v_0,v_l)权重为0的。

从v_0出发，运行bellman-ford，若存在一条负权重和的环路，则判断存在此种序列，反之，不存在此种序列。

运行时间：创建具有$O(n^2)$条边的有向图G需要$O(n^2)$，需要$O(n^3)$执行bellman-ford算法，故总时间为$O(n^3)$。

- b.假设bellman-ford算法执行结果存在false返回值，即a中序列存在。此时，根据结点的前驱π可以找到负权重环路上的一个结点，然后沿环路继续搜索，直到到达一个我们之前曾经访问过的结点x。该回路就是我们要找的回路。
- 可以使用22.2节中打印路径的算法，在返回至结点x时，打印路径终止即可。
- 时间分析：花费$O(n^3)$执行bellman-ford，花$O(n)$时间打印环路中节点，总共$O(n^3)$时间。

25.2-6 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？

方法1：

检测结果矩阵主对角线是否存在负值。

方法2：

多执行一遍循环，如果还有矩阵元素更新，这表明有负权重的环路。