1、给定一个赋权有向图$D=(V,A)$，记D中每一条 弧$a=(v_i,v_j)$上的权为w_{ij}，D就成为赋权有向图（网络图）. 给定D中一个起点v_s和终点v_t，设P是D中从v_s到v_t的一条路. 把P中所有弧的权之和称为路P的权. 记为$W(P)$。 即

$$W(P)=\sum_{(v_i,v_j)} w_{ij}$$

又若p^*是D图中从v_s到v_t的一条路，且满足

$$w(p^*)=\min\{w(p)|p\text{ 为}v_s\text{到}v_t\text{的路}\}$$

式中对D的所有从v_s到v_t的路p取最小，则称p^*为从v_s到v_t的最短路，$w(p^*)$为从v_s到v_t的最短距离。记为$d(v_i,v_j)$。在一个图$D=(V,A)$中，求从v_s到v_t的最短路和最短距离的问题就称为最短路问题。

这里说的“距离只是权数”的代称，在实际的网络图中，权数也可以是时间、费用等等；

2、求最短路有两种算法：

一是求从**某一点至其他各点**之间最短距离的狄克斯屈拉（Dijkstra）算法；

另一种是求网络图上**任意两点之间最短距离**的矩阵算法福德（Ford）算法；

3、Dijkstra 算法实际上也给出了寻求从一个始定点v_s到任意一个点v_j的最短路。目前认为，在所有权$w_{ij}\geq 0$时，这个算法是寻求最短路问题最好的算法。

Dijkstra 算法基于的事实：如果P是D中从v_s到v_j的最短路，v_i是P中的一点，那么从v_s沿P到v_i路也是从v_s到v_i的最短路。

4、Dijkstra 算法的基本思路：

（1）求从图的一点到其它各点之间最短路

（2）从始点出发，逐步顺序地向外探寻

（3）每向外延伸一步都要求是最短的

Dijkstra 算法的标号说明：

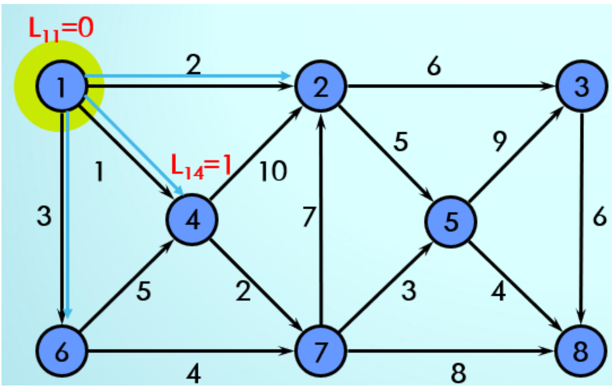
（1）用d_{ij}表示图中两相邻点i与j的距离，若i与j不相邻，令$d_{ij}=\infty$，显然$d_{ii}=0$

（2）用L_{si}表示从s点到i点的最短距离，标注于i点旁，并称i点为标号点。

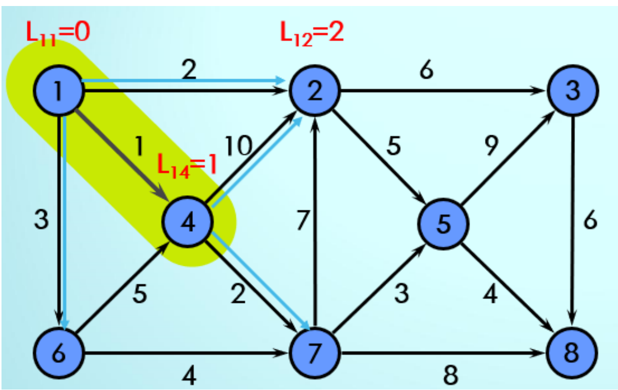
（3）当有中间点k时，$L_{si}=L_{sk}+d_{ki}$

（4）标号的点表示已找到它与始点之间的最短路

5、Dijkstra 算法例题

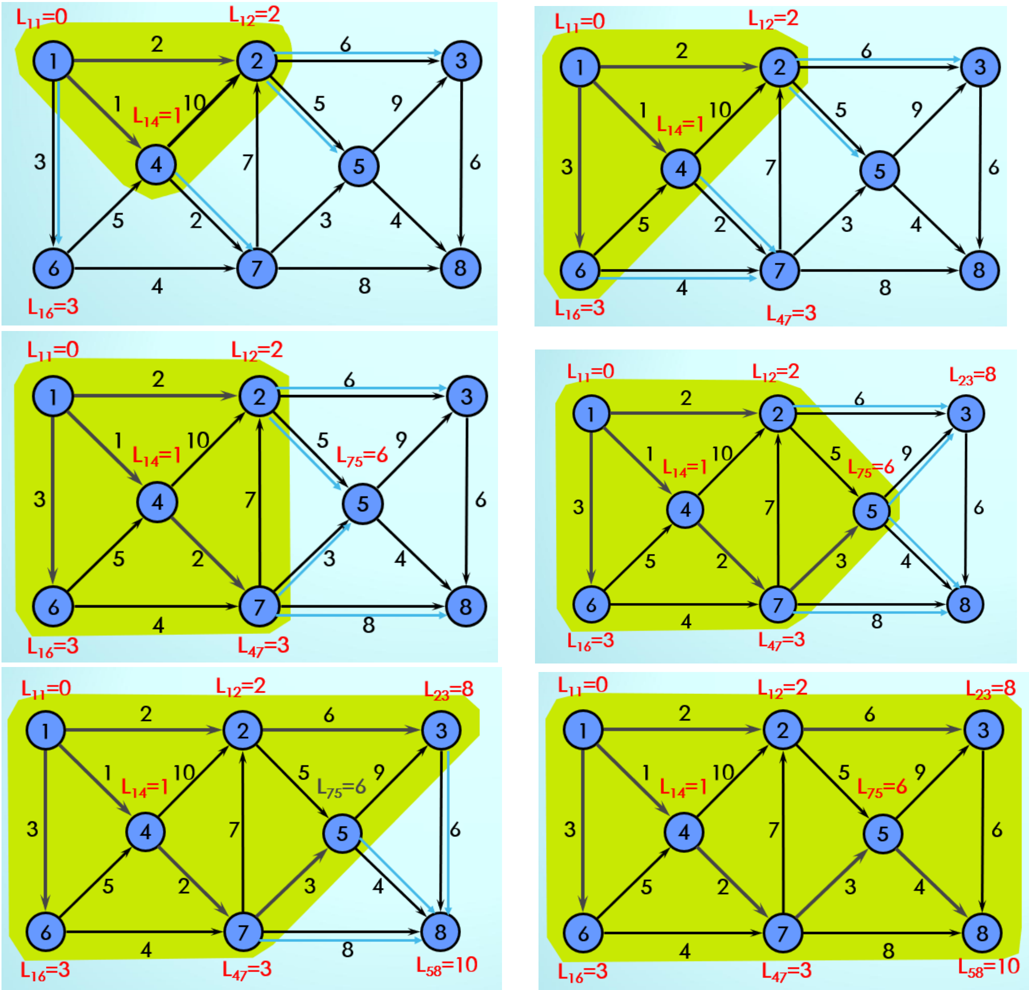


1. 从点 s 出发，因 Lss=0，将此值标注在 s 旁，表示 s 点已标号；
2. 从 s 点出发，找出与 s 相邻的点中距离最小的一个，设为 r。将 Lsr=Lss+dsr 的值标注在 r 旁，表明点 r 也已标号；
此为有向图，注意不可逆向



3. 已标号的点出发，找出与这些点相邻的所有未标号点 p。若有 Lsp=min{Lss+dsp, Lsr+drp}，则对 p 点标号，并将 Lsp 的值标注在 p 点旁

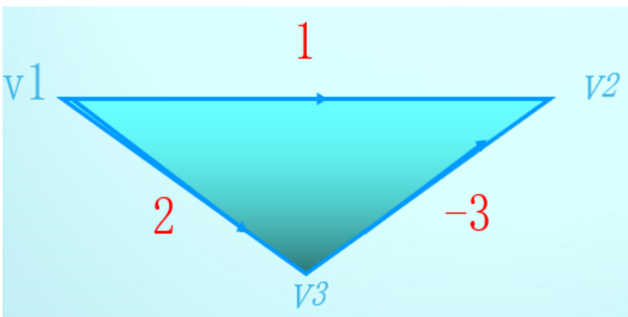
4. 重复第 3 步，一直到 t 点得到标号为止



1 到 8 的最短路径为{1，4，7，5，8}，长度为 10，1 到所有点的最短路机器距离都以求出并标出。

6、Dijkstra 算法局限性

Dijkstra 算法仅适用于所有的权$w_{ij} \geq 0$的情形，如果当赋权有向图中存在负权弧时，则该算法失效。例如下图中，



根据 Dijkstra 算法，可以得到从v_1到v_2最短路的权是 1，但是这显然不对，因为从v_1到v_2的最短路是(v_1, v_3, v_2)，权是−1

另外，Dijkstra 算法只能用来求解从某一点至其它各点之间最短距离，而不能用来求解网络图上任意两点间的最短距离。

7、最短路的 Floyd（弗洛伊德）算法

Floyd 算法时更一般的算法，该算法是一种矩阵(表格)迭代的方法，对于求任意两点间的最短路，混合最短路，有赋权图的最短路等一般网络问题来说比较有效。

下面给出有负权数的，但无负回路的有向（或无向）网络的最短路问题的方法：

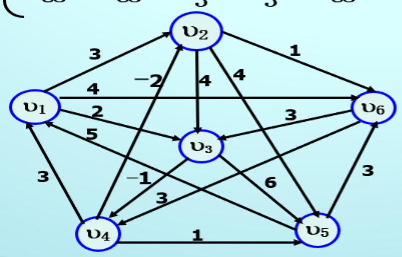
首先，设从任一点v_i到任一点v_j都有一条弧，如果在图 D 中，(v_i,v_j)不是弧，则添加弧 (v_i,v_j)，并且令$w_{ij}=+\infty$. 很明显，从v_s到v_j的最短路是从v_s点出发，沿着这条路到某个点v_i的再沿弧(v_i,v_j)到点v_j。显然，从v_s到v_i的这条路必定是从v_s到v_i的最短路。否则从v_s到v_j的这条路将不是最短路。于是，从v_s到v_j的距离$d(v_s,v_j)$满足以下条件：

$$d(v_s,v_j)=\min\{d(v_s,v_i)+w_{ij}\}\quad ,i=1,\dots,p,p=p(D)$$

8、Floyd（弗洛伊德）算法例题

$$\mathbf{W}=\begin{array}{c} \text{从\\至} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}\begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 0 & 3 & 2 & \infty & \infty & 4 \\ \infty & 0 & 4 & \infty & 4 & 1 \\ \infty & \infty & 0 & -1 & 6 & \infty \\ 3 & -2 & \infty & 0 & 1 & \infty \\ 5 & \infty & \infty & \infty & 0 & 3 \\ \infty & \infty & 3 & 3 & \infty & 0 \end{pmatrix} \end{array}$$

在其中一定要注意从\至关系



$$w*d_1=\begin{bmatrix}0&3&2&\infty&\infty&4\\\infty&0&4&\infty&4&1\\\infty&\infty&0&-1&6&\infty\\3&-2&\infty&0&1&\infty\\5&\infty&\infty&\infty&0&3\\\infty&\infty&3&3&\infty&0\end{bmatrix}*\begin{bmatrix}4\\1\\\infty\\\infty\\3\\0\end{bmatrix}\to\begin{bmatrix}4\\1\\9\\-1\\3\\0\end{bmatrix}\to\begin{bmatrix}4\\1\\-2\\-1\\3\\0\end{bmatrix}\to\begin{bmatrix}0\\1\\-2\\-1\\3\\0\end{bmatrix}$$

$$\to\begin{bmatrix}0\\1\\-2\\-1\\3\\0\end{bmatrix}\to\begin{bmatrix}0\\1\\-2\\-1\\3\\0\end{bmatrix}$$

根据下式：

$$w*d_1=\begin{bmatrix}0&3&2&\infty&\infty&4\\\infty&0&4&\infty&4&1\\\infty&\infty&0&-1&6&\infty\\3&-2&\infty&0&1&\infty\\5&\infty&\infty&\infty&0&3\\\infty&\infty&3&3&\infty&0\end{bmatrix}*\begin{bmatrix}4\\1\\\infty\\\infty\\3\\0\end{bmatrix}\to\begin{bmatrix}0\\1\\-2\\-1\\3\\0\end{bmatrix}\to\begin{bmatrix}0\\1\\-2\\-1\\3\\0\end{bmatrix}$$

找到是矩阵中的哪两个数相乘得到的最终结果。

最短路：按 W 阵中画圈元素的从\至关系：

$v_1\to v_3\to v_4\to v_2\to v_6$　　2−1−2+1=0

$v_2\to v_6$

$v_3\to v_4\to v_2\to v_6$

$v_4\to v_2\to v_6$

$v_5\to v_6$

$v_6\to v_6$

9、至\从关系

求"某点至"各点的最短路

$$
l_1^T * W = (0 \quad 3 \quad 2 \quad \infty \quad \infty \quad 4) *
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \text{至}\backslash\text{从} \end{array} \\
\left(\begin{array}{cccccc}
0 & 3 & 2 & \infty & \infty & 4 \\
\infty & 0 & 4 & \infty & 4 & 1 \\
\infty & \infty & 0 & -1 & 6 & \infty \\
3 & -2 & \infty & 0 & 1 & \infty \\
5 & \infty & \infty & \infty & 0 & 3 \\
\infty & \infty & 3 & 3 & \infty & 0
\end{array}\right)
\end{array}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
$$

$\Rightarrow l_2^T = (0 \quad 3 \quad 2 \quad 1 \quad 7 \quad 4)$

$\Rightarrow l_3^T = (0 \quad -1 \quad 2 \quad 1 \quad 2 \quad 4)$

$\Rightarrow l_4^T = (0 \quad -1 \quad 2 \quad 1 \quad 2 \quad 0)$

$\Rightarrow l_5^T = (0 \quad -1 \quad 2 \quad 1 \quad 2 \quad 0)$

最短路	最短路长
$v_1 \to v_1$	0
$v_1 \to v_3 \to v_4 \to v_2$	−1
$v_1 \to v_3$	2
$v_1 \to v_3 \to v_4$	1
$v_1 \to v_3 \to v_4 \to v_5$	2
$v_1 \to v_3 \to v_4 \to v_2 \to v_6$	0

10、记矩阵 $D_k = (d_{ij}{}^{(k)})_{n\times n}$，当 $k=1$ 时，令 $D_1 = (d_{ij}{}^{(1)})_{n\times n} = (w_{ij})_{n\times n} = w$，定义 $D_k = D_{k-1} * D_{k-1}, k = 2,3,\dots,p$，其中 $d_{ij}{}^{(k)} = \min\{d_{is}{}^{(k-1)} + d_{sj}{}^{(k-1)}\}, i,j = 1,2,\dots,n$ 若 $w_{ij} \geq 0, i,j = 1,2,\dots,n$，则关于 p 值有下述估式：$2^{p-1} \leq n-1 \leq 2^p$，则

$p \geq \lg(n-1)/lg2 \geq p-1$，也可以计算到 $D_k = D_{k-1}, k = 2,3,\dots$ 时停止计算。

11、网络规划解决设备更新问题

各年初购价					
年 度 i	1	2	3	4	5
年初购价p_i	13	14	16	19	24

各年维护费					
使用年数 k	1	2	3	4	5
第k年维护费c_k	8	10	13	18	27

解：设以

v_i 表示第 i 年初这一状态，i=1, 2, ⋯, 5

v_6表示第 5 年末这一状态

(v_i, v_j)第 i 年初购置的一台新设备一直使用到第 j 年初这一方案

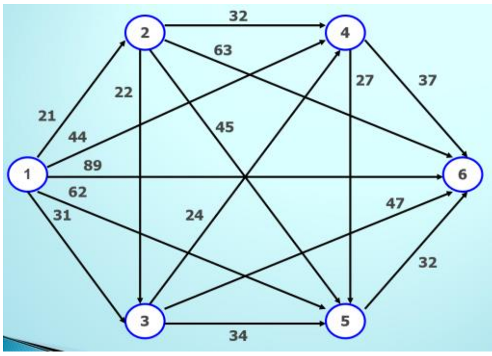
w_{ij}方案(v_i, v_j)的费用

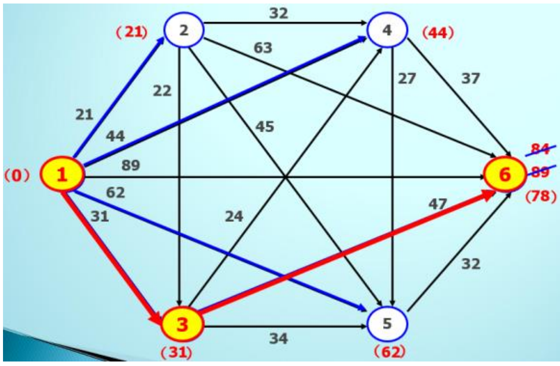
则

$$w_{ij} = p_i + \sum_{k=1}^{j-i} c_k$$

累积维护费

使用年数 j−i	1	2	3	4	5
第年维护费c_k	8	10	13	18	27
累积维护费$\sum c_k$	8	18	31	49	76





12、网络的中心和重心

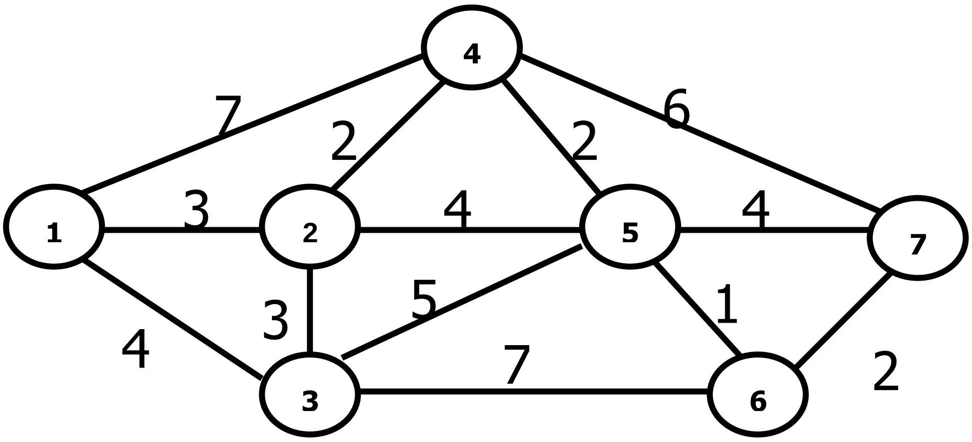
（1）网络的中心

设 $D = (d_{ij})$ 为网络各点间最短距离矩阵。令 $d(v_i) = \max\{d_{ij}\}, i = 1,2,\dots,n\ \ 1 \le j \le n$，若$\min\{d(v_i)\} = d(v_k)$，则$v_k$称为网络的中心.

（2）网络的重心

设 g_i 为点 v_i 的权重（i=1, 2, ⋯, n），令 $h(v_i) = \sum_{i=1}^{n} g_i\, d_{ij}, j = 1,2,\dots,n$，若$\min\{h(v_i)\} = h(v_r)$，则称$v_r$为网络的重心

例：某地 7 个村镇之间的现有交通道路如图所示，边旁数字为各村镇之间道路的长度。现要沿交通道路架设电话线，使各村之间均能通话。应如何架线使费用最省？



(1) 商店应建在何村，能使各村都离它较近？

(2) 小学应建在何村，能使各村小学生到校总里程最小？

村 镇	1	2	3	4	5	6	7
小学生人数	40	25	45	30	20	25	50

(1) 商店应建在v_4村（中心）

υ_j / υ_i	D=(d_{ij})							$d(\upsilon_i)=\max\{d_{ij}\}$
	1	2	3	4	5	6	7	
1	0	3	4	5	7	8	10	10
2	3	0	3	2	4	5	7	7
3	4	3	0	5	5	6	8	8
$\boxed{4}$	5	2	5	0	2	3	5	5 min
5	7	4	5	2	0	1	3	7
6	8	5	6	3	1	0	2	8
7	10	7	8	5	3	2	0	10

(2) 小学应建在v_5村(重心)

g_i	υ_i \ υ_j	$g_i \times d_{ij}$						
		1	2	3	4	$\boxed{5}$	6	7
40	1	0 0	3 120	4 160	5 200	7 280	8 320	10 400
25	2	3 75	0 0	3 75	2 50	4 100	5 125	7 175
45	3	4 180	3 135	0 0	5 225	5 225	6 270	8 360
30	4	5 150	2 60	5 150	0 0	2 60	3 90	5 150
20	5	7 140	4 80	5 100	2 40	0 0	1 20	3 60
35	6	8 280	5 175	6 210	3 105	1 35	0 0	2 70
50	7	10 500	7 350	8 400	5 250	3 150	2 100	0 0
	$h(\upsilon_j)$	1325	920	1095	870	850	925	1215