第一章 图的基本概念

本次课主要内容

(一)、最短路算法

(二)、图的代数表示

1、图的邻接矩阵

2、图的关联矩阵

(一)、最短路算法

1、相关概念

(1) 赋权图

在图G的每条边上标上一个实数w (e)后, 称G为边赋权图。被标上的实数称为边的权值。

若H是赋权图G的一个子图，称 W(H)=$\sum_{e\in E(H)} w(e)$ 为子图H的权值。

权值的意义是广泛的。可以表示距离，可以表示交通运费，可以表示网络流量，在朋友关系图甚至可以表示友谊深度。但都可以抽象为距离。

(2) 赋权图中的最短路

设G为边赋权图。u与v是G中两点，在连接u与v的所有路中，路中各边权值之和最小的路，称为u与v间的最短路。

(3) 算法

解决某类问题的一组有穷规则，称为算法。

1) 好算法

算法总运算量是问题规模的多项式函数时，称该算法为好算法，也称有效算法。有好算法的问题称为P类问题

问题规模：描述或表示问题需要的信息量。 通常用图的顶点数和边数描述。

算法中的运算包括算术运算、比较运算等。运算量用运算次数表示。

2) 算法分析

对算法进行分析，主要对时间复杂性进行分析。分析运算量和问题规模之间的关系。

最短路算法

- **问题：** 若干个城市被铁路网连通， 任意指定其中的两座城市， 试求这两座城市之间的最近的铁路路线。

- **图论模型：** 城市作为顶点， 仅当两城市有一段铁路，而这段铁路中途没有其他火车站，则将这两个城市连边，如此构成一个图。 对每条边e赋权w(e), w(e)代表e的长度，得到加权的连通图。 Π(u, v)代表以u，v为端点的线路的集合， P（u，v)表示u到v的一条线路，

W（P(u, v）） 代表P（u， v）上所有边的权之和，

即 W(P(u, v))=$\sum_{e\in P(u,v)} w(e)$

目标是求一条线路 $W(P_0(u,v)) = \min_{p(u,v)\in\Pi(u,v)} \{W(P(u,v))\}$

Dijkstra算法

Dijkstra算法能求一个顶点到另一顶点最短路径。它是由Dijkstra于1959年提出的，也称狄克斯特拉算法。实际它能求出始点到其它所有顶点的最短路径。

Dijkstra算法基本思想：

它是一种标号法：给赋权图的每一个顶点记一个数，称为顶点的标号（固定标号（S_i中的点）或者临时标号（ $V-S_i$中的点））；临时标号表示从始顶点到该标点的最短路长的上界；固定标号则是从始顶点到该顶点的最短路长。

Dijkstra 算法：

(1) u, v 不相邻时，取 $w(uv)=\infty$.

(2) 令 $l(u_0)=0$; $l(v)=\infty$, $v\neq u_0$; $S_0=\{u_0\}$, $i=0$.

(3) 对每个 $v\notin S_i$，用

$$\min\{l(v), l(u_i)+w(u_i v)\}$$

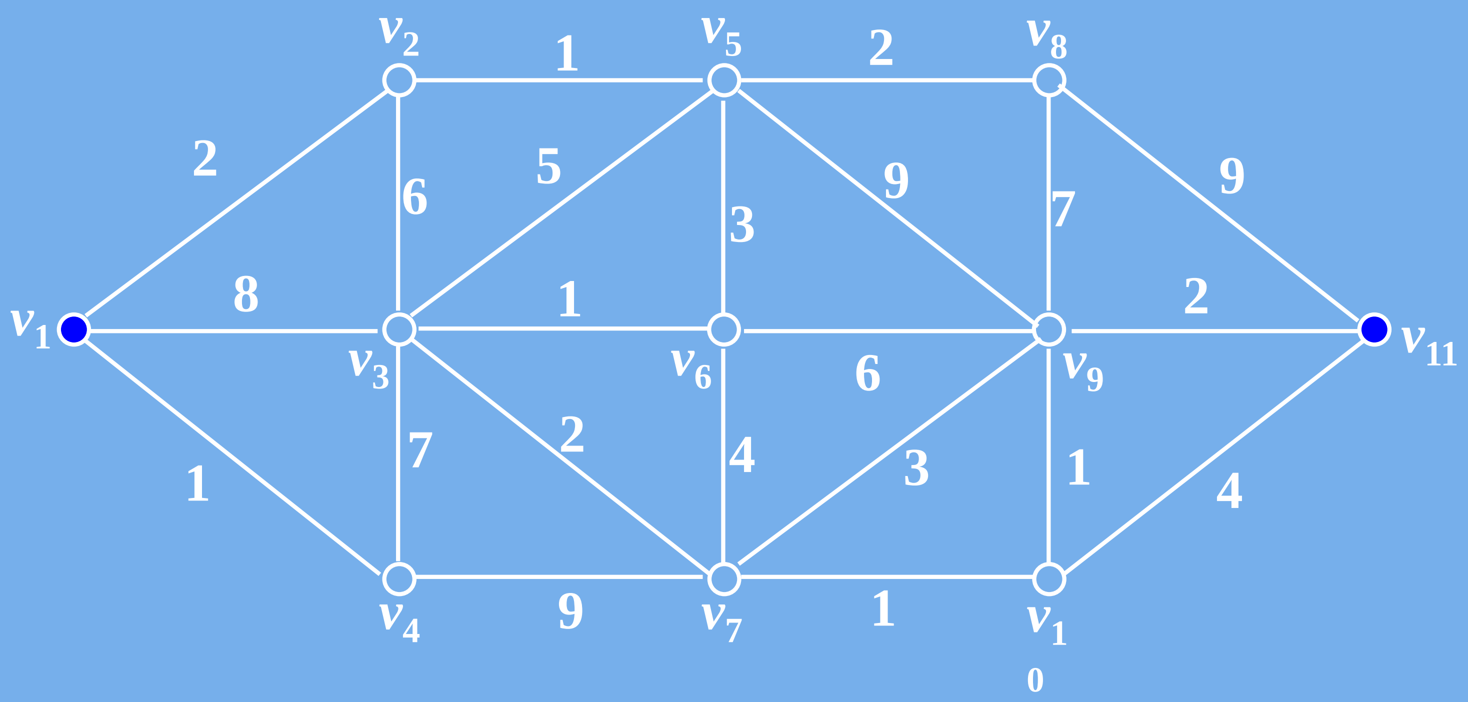
替代 $l(v)$；设 u_{i+1} 是使 $l(v)$ 取最小值的 $V(G)-S_i$ 中的顶，令 $S_{i+1}=S_i\cup\{u_{i+1}\}$.

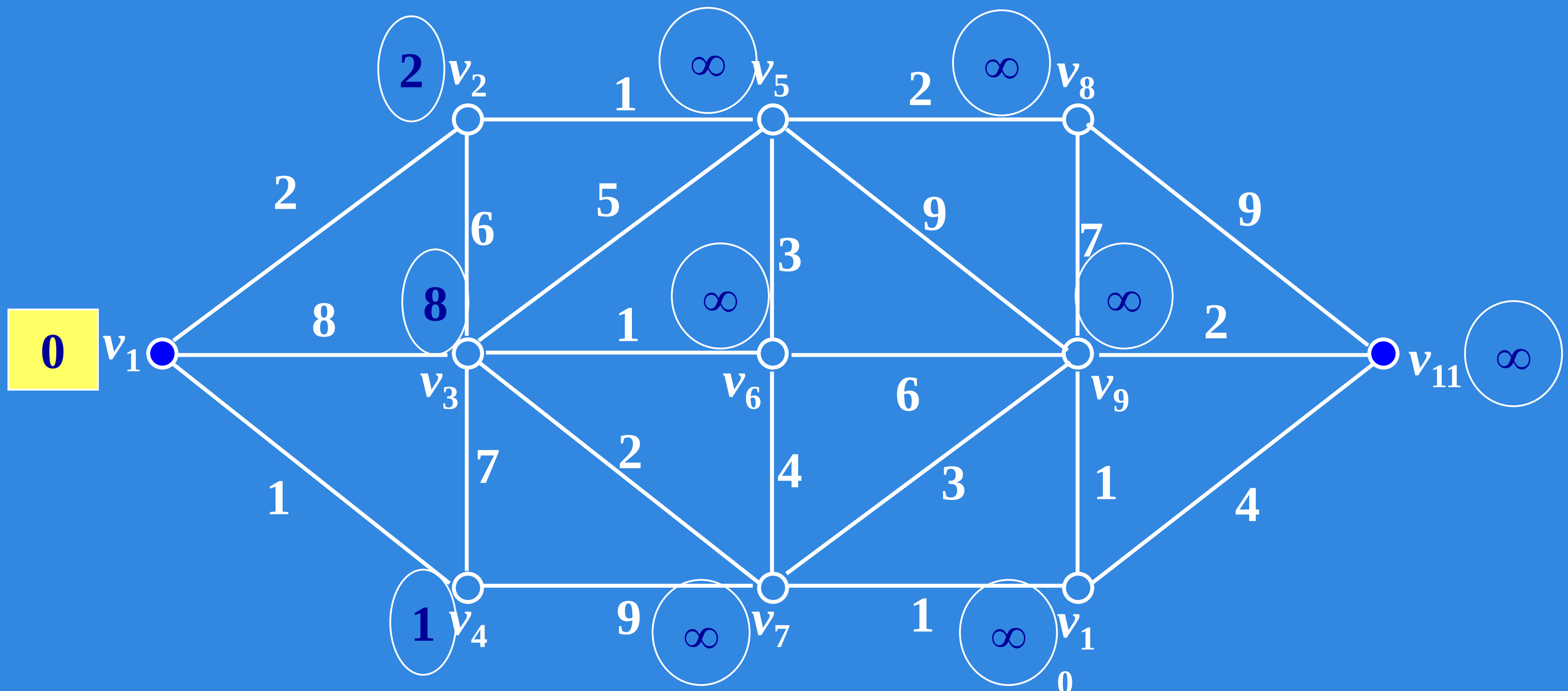
(4) $i=\nu-1$，止；若 $i<\nu-1$，用 $i+1$ 代替 i，转(3).

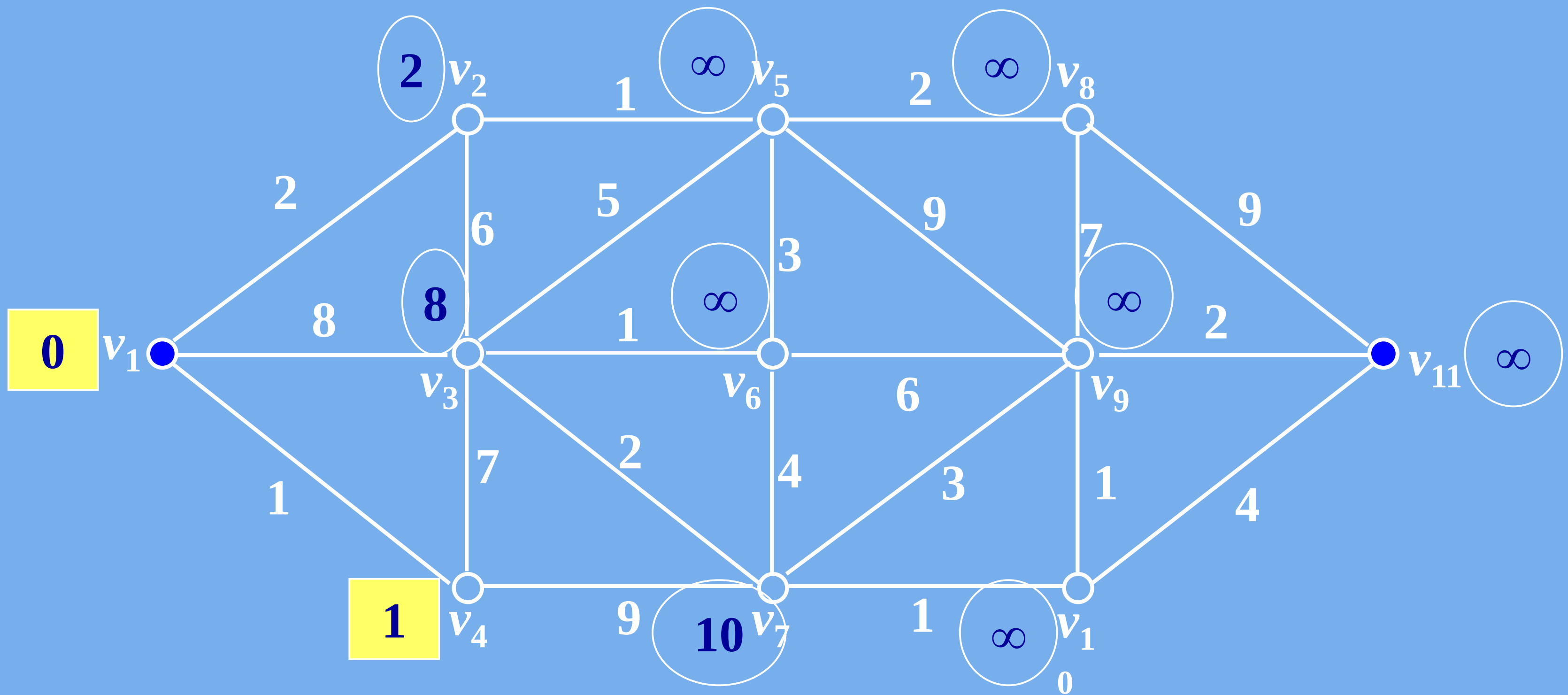
容易看出S中的元的标号是固定标号，**l(u)**代表起点到u的距离，V-S中的标号是临时标号，**l(u)**代表起点到u的距离的上界

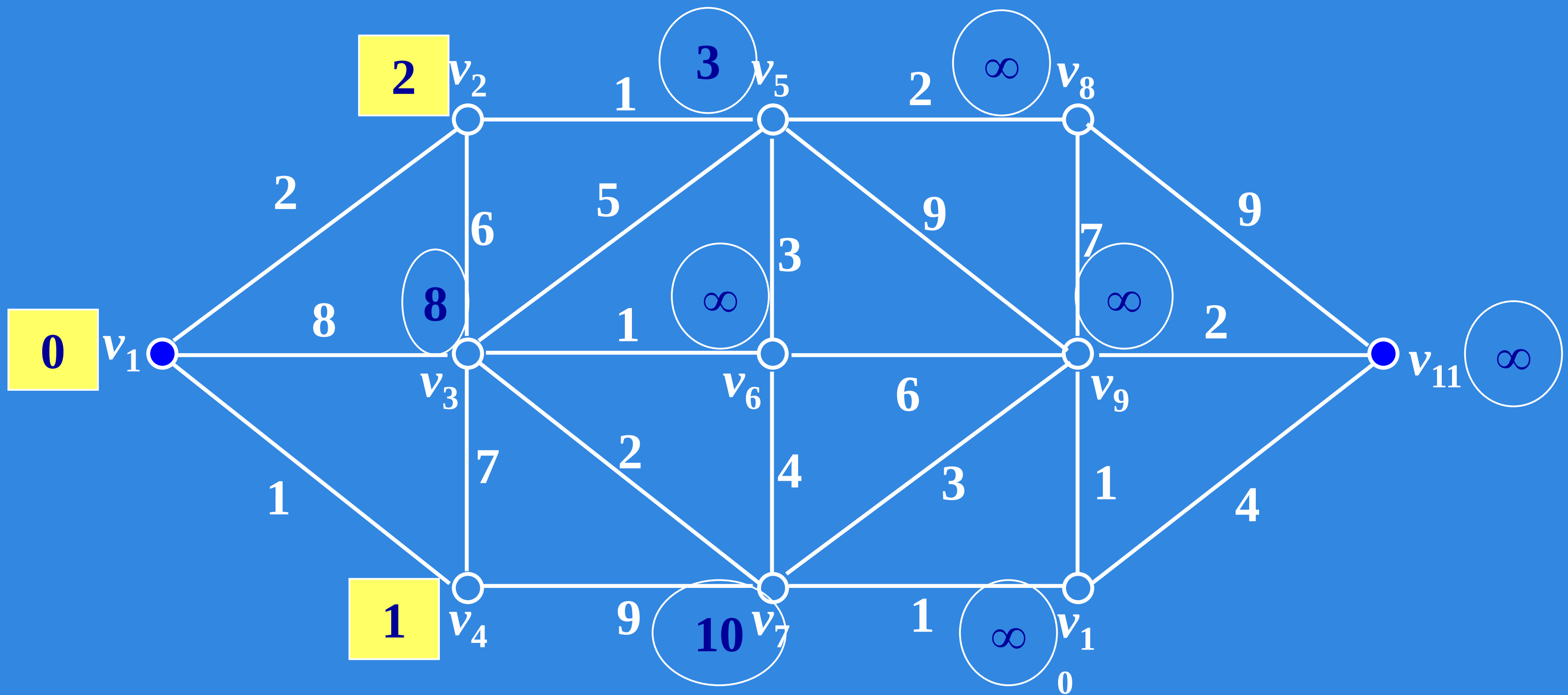
- 由于G是有限图，故有限步之后，可以逐次求出起点到每个顶的距离，而且可以在算法执行中标志出起点到各个点的一条最短路。
- Dijkstra算法的时间复杂度是O(n^2),是有效算法。
- 下面例子中顶点旁的黄色方框里的数字表示固定标号，顶点旁的圈圈里的数字表示临时标号

算法举例

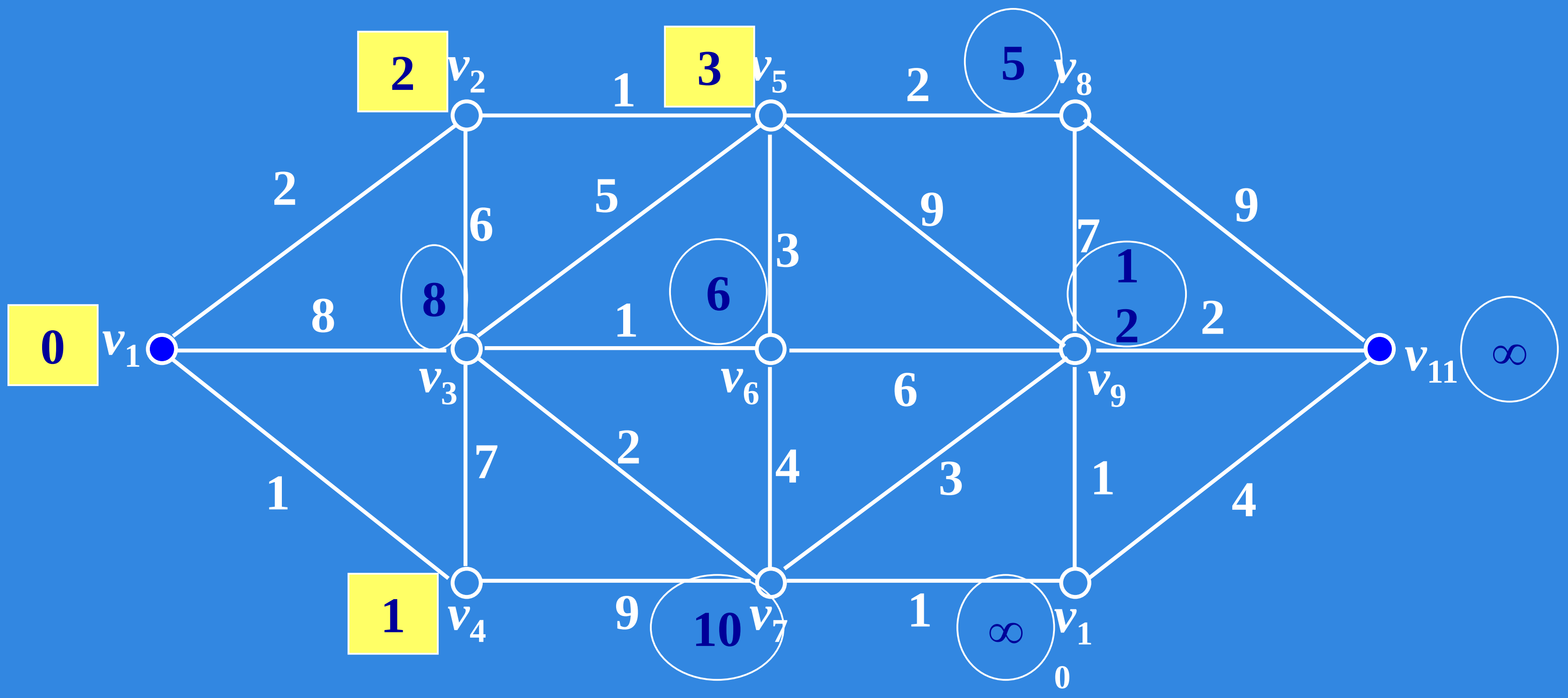



0
v1
2
v2
1
∞
v5
2
∞
v8
2
6
5
9
9
3
7
8
8
1
∞
∞
2
∞
v3
v6
6
v9
v11
1
7
2
4
3
1
4
9
1
1
v4
∞
v7
1
∞
v1
0

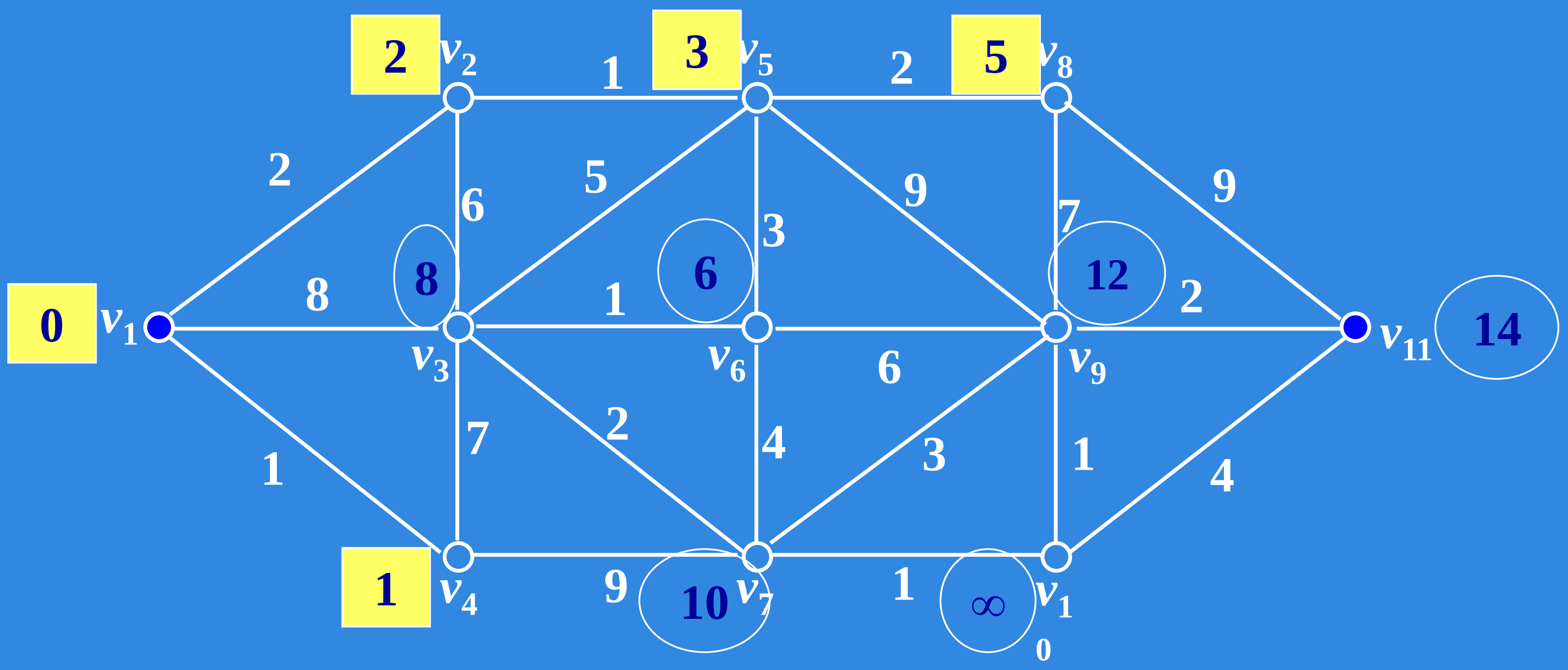

0
1
2
8
10
∞
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}
2
8
1
1
6
7
5
1
2
9
3
4
2
9
7
2
6
3
1
9
1
4



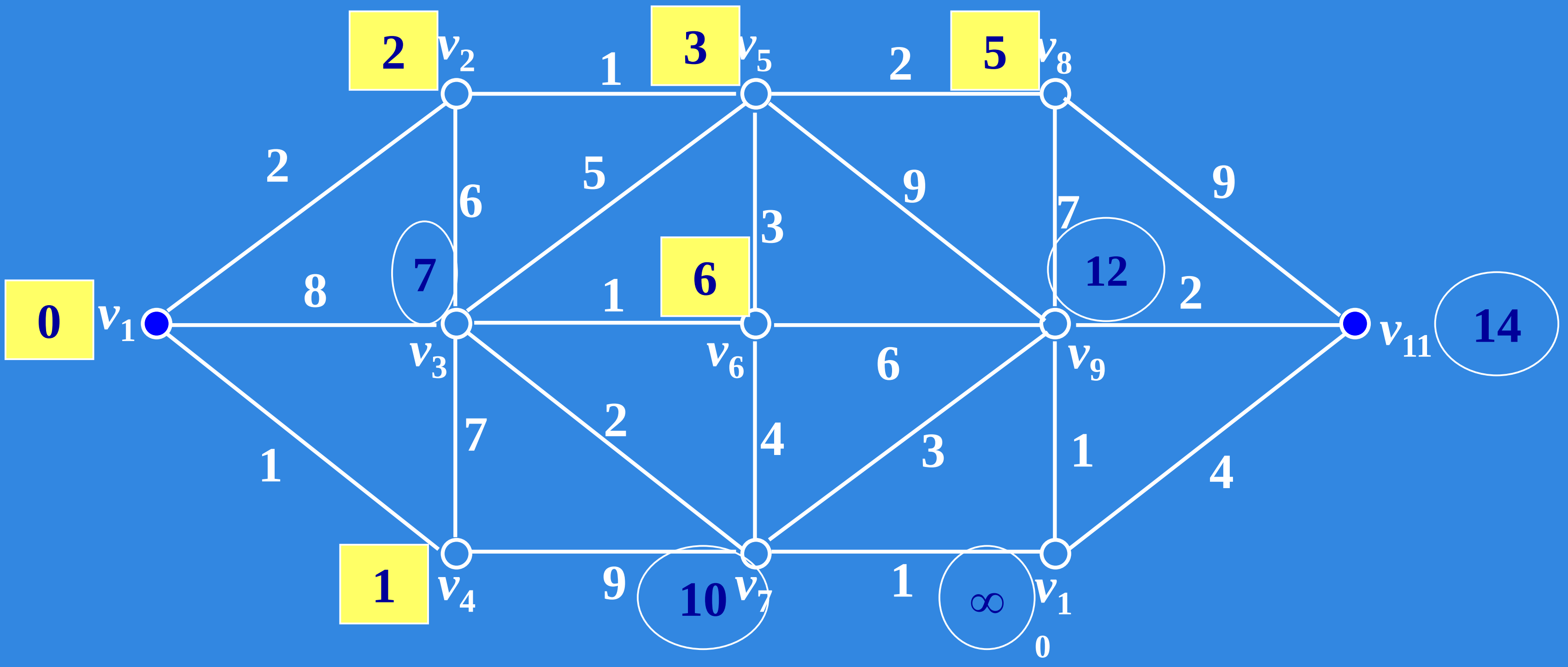
2
v_2
1
3
v_5
2
∞
v_8
2
6
5
3
9
7
9
0
v_1
8
8
1
∞
∞
2
v_{11}
∞
v_3
v_6
6
v_9
7
2
4
3
1
1
4
1
v_4
9
10
v_7
1
∞
v_{10}



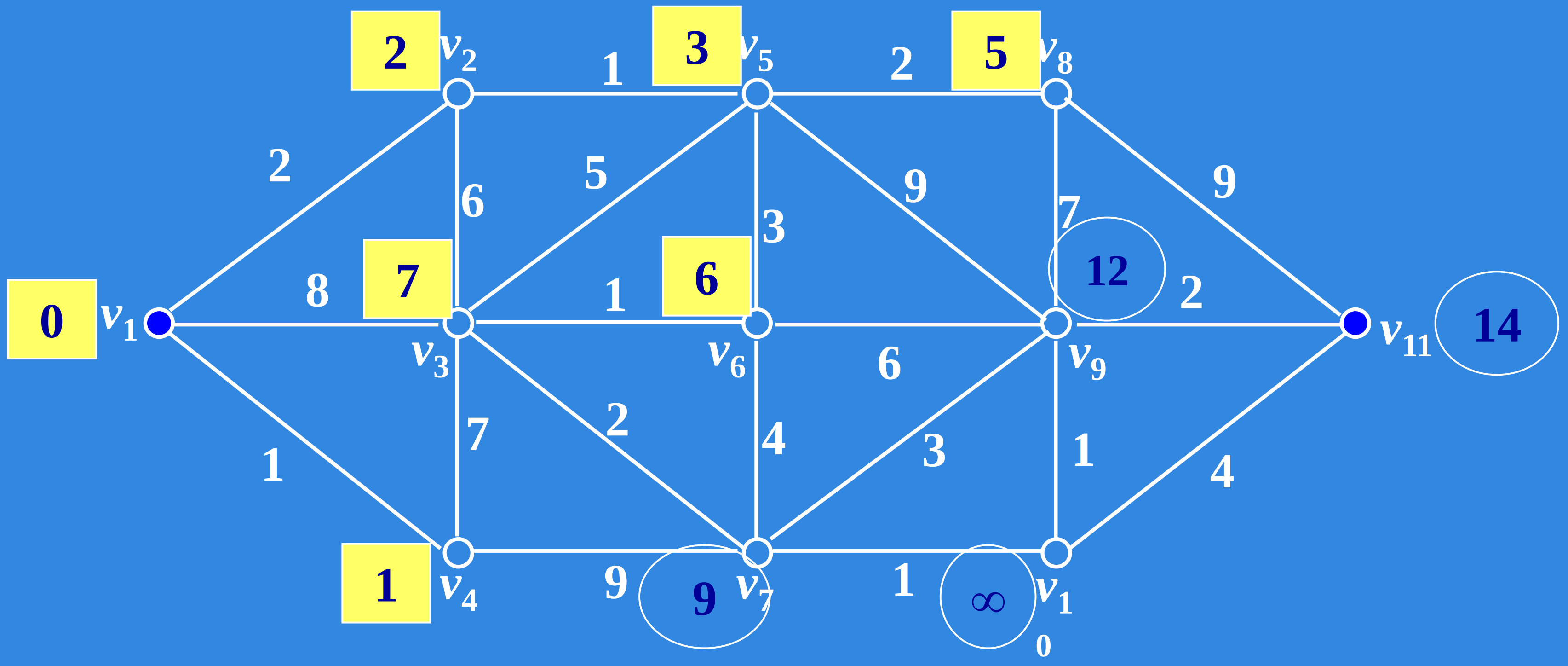
0
2
3
5
8
6
12
1
10
∞
∞
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11
2
8
1
1
6
7
5
1
2
9
2
9
3
4
3
1
7
6
1
9
2
4



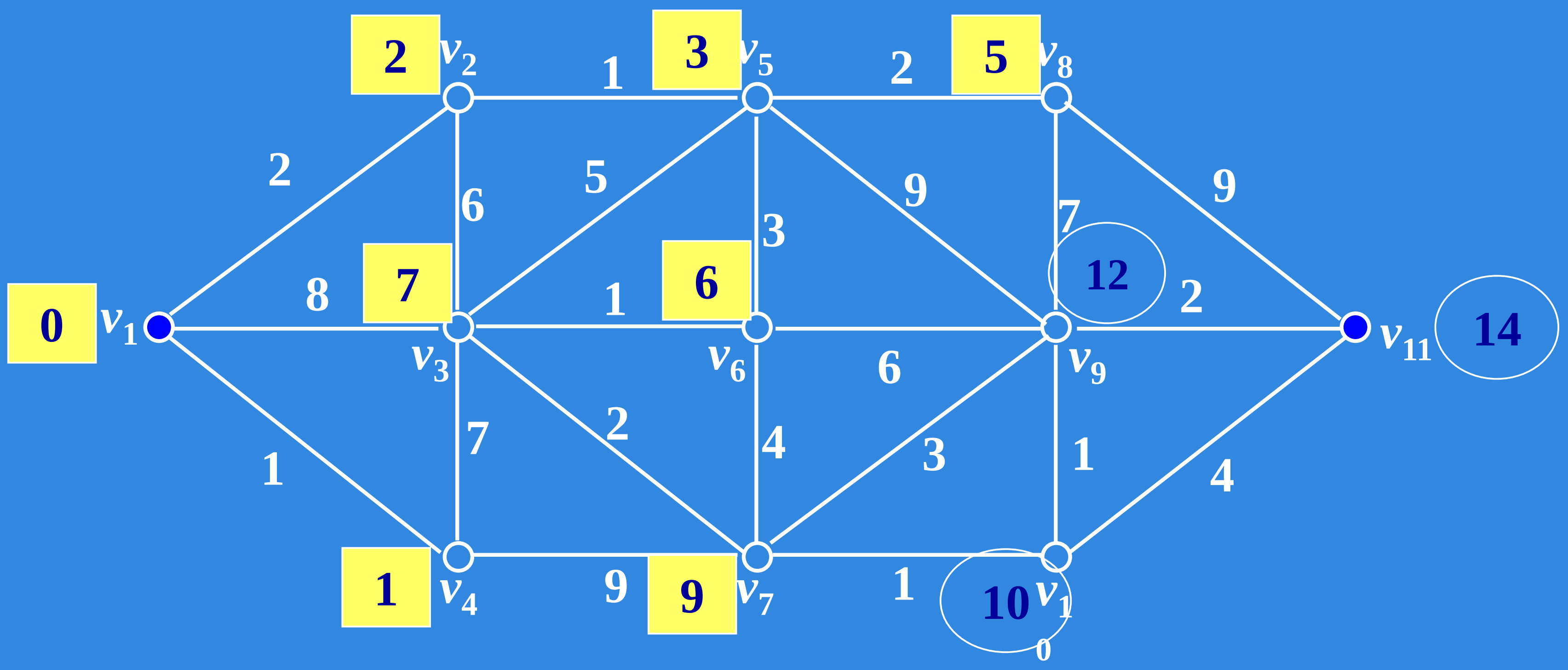
2
v2
1
3
v5
2
5
v8
2
5
6
3
9
7
9
8
6
12
2
0
v1
8
1
v3
v6
6
v9
v11
14
1
7
2
4
3
1
4
1
v4
9
10
v7
1
∞
v10



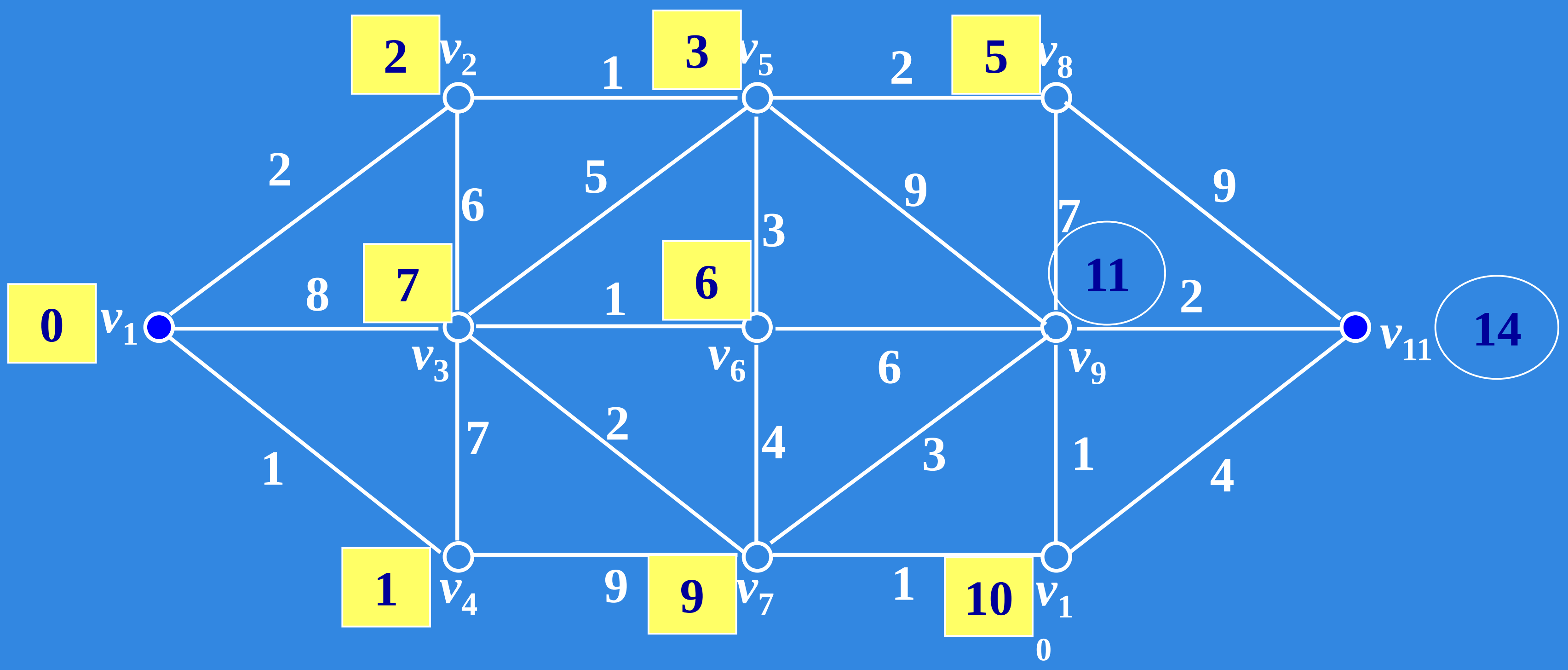
0
v_1
2
v_2
1
3
v_5
2
5
v_8
2
6
5
8
7
v_3
1
6
v_6
3
9
7
12
9
2
v_9
v_{11}
14
6
1
7
2
4
3
1
4
1
v_4
9
10
v_7
1
∞
v_{10}



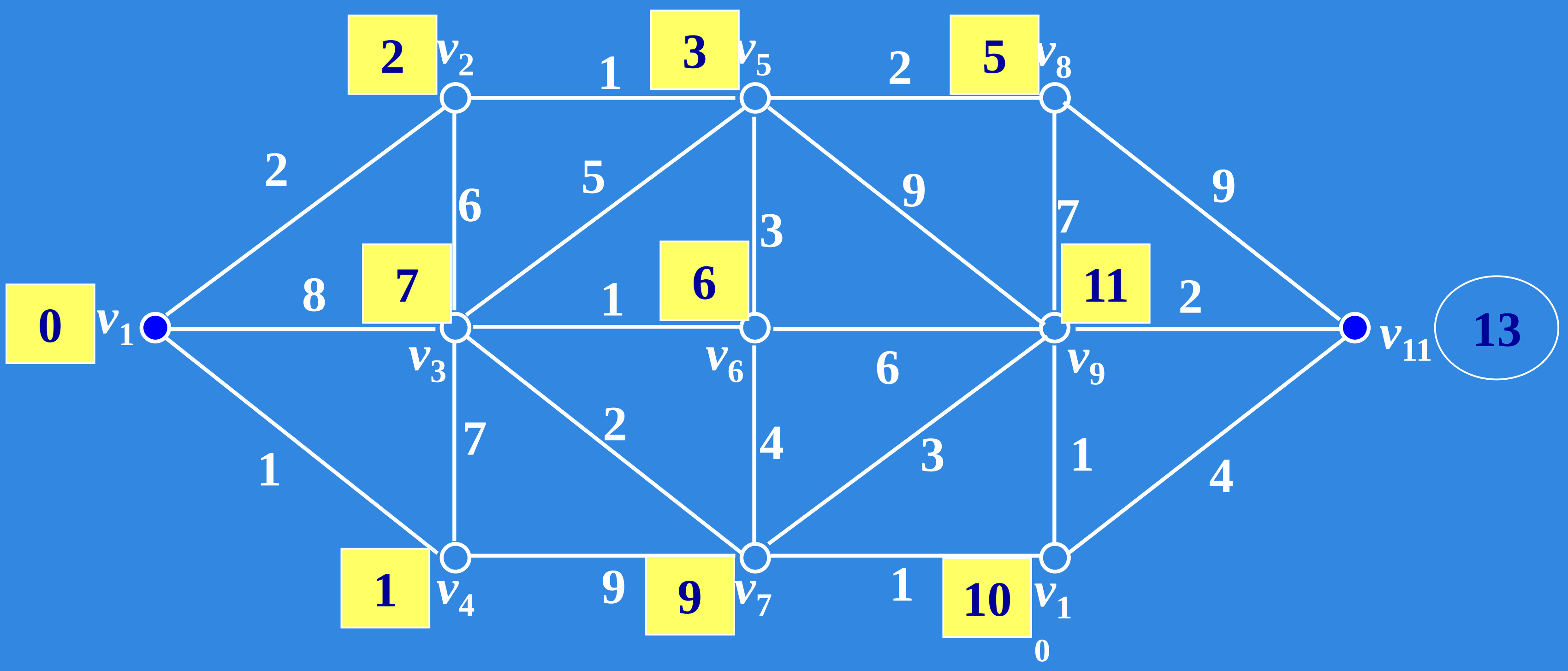
0
2
7
1
3
6
5
9
∞
12
14
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}
2
8
1
1
6
7
5
1
2
9
3
4
2
9
6
3
1
7
1
9
2
4



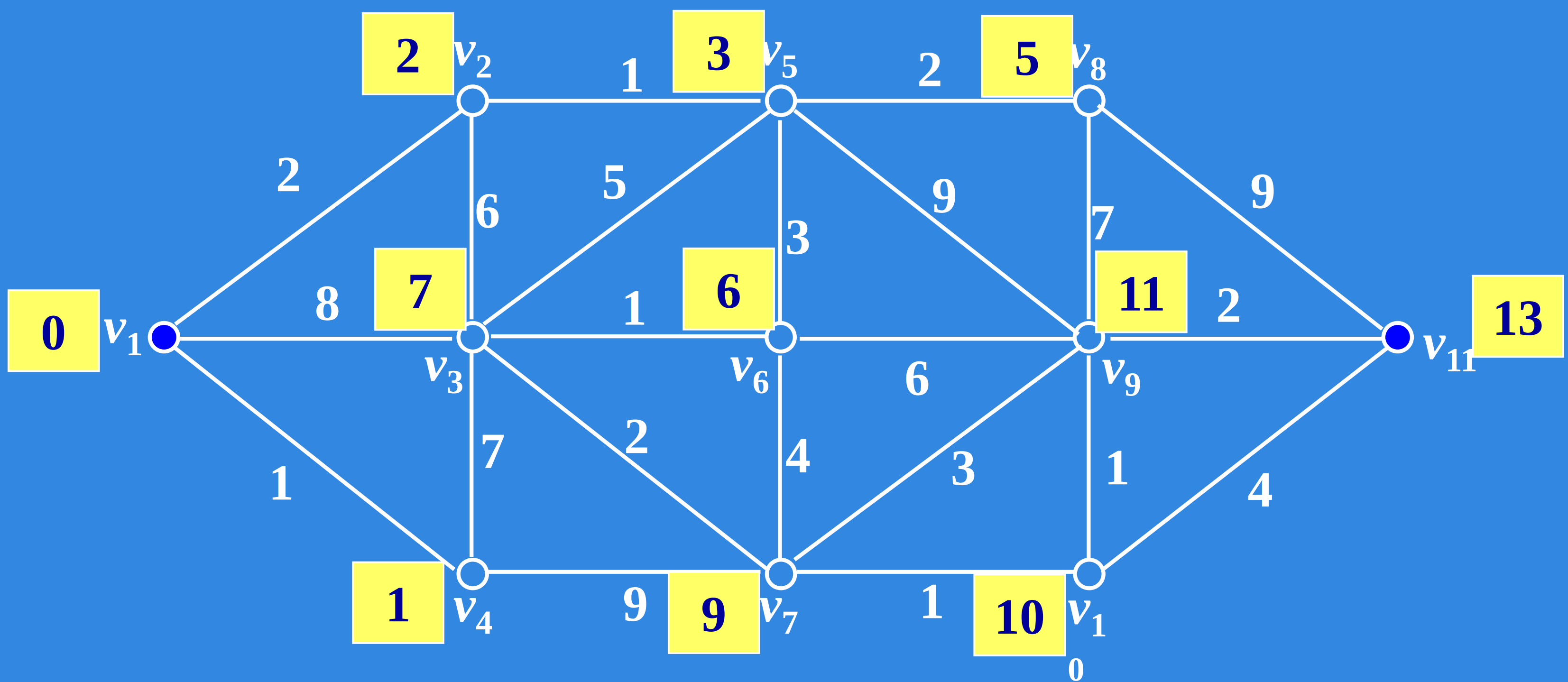
0
v1
2
v2
1
3
v5
2
5
v8
2
6
5
9
3
7
9
8
7
1
6
12
2
v3
v6
6
v9
v11
14
1
7
2
4
3
1
4
1
v4
9
9
v7
1
10
v10



0
2
7
1
3
6
9
5
10
11
14
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}



0
v1
2
v2
1
3
v5
2
5
v8
2
6
5
9
3
7
9
8
7
1
6
11
2
v3
v6
6
v9
v11
13
1
7
2
4
3
1
4
1
v4
9
9
v7
1
10
v10



2
v_2
1
3
v_5
2
5
v_8
2
5
6
9
3
7
9
0
v_1
8
7
1
6
11
2
13
v_3
v_6
6
v_9
v_{11}
7
2
4
3
1
1
4
1
v_4
9
9
v_7
1
10
v_{10}

- Dijkstra算法正确性的证明？？？

算法的正确性是显然的。 因为在任意一步， 设S中每一点的l标号是u_0到该点的最短路长（开始$S=\{u_0\}$， $l(u_0)=0$，这个假设显然是对的）， 假设$S=\{u_0, u_1, \cdots, u_i\}$， 其中每个点的$l(u_j)$都是$u_0$到$u_j$的最短距离， $j=0, 1, \cdots, i$.

那么只要说明$l(u_{i+1})$是u_0到u_{i+1}的最短路长就行了。

反证，假设加入的u_{i+1}不是最短路径，即存在另外一条路径为最短路径，设u_k为该路径中不在集合S的第一个结点，则$l(u_k)<l(u_{i+1})$，dijkstra的算法的第一个步骤是每次从集合V-S中基于l选取最小的结点加入集合S，因此 $l(u_{i+1})\leqslant l(u_k)$ ，因此产生矛盾。

Floyd算法

- Dijkstra算法只求出图中一个特定点到其他各点的最短路，而在许多实际问题中需要求出任意两点之间的最短路，如全国各个城市之间的最短航线，选址问题等。

其实要求出一个图中任意两点间的最短路，只需将图中每个点依次视为起点，然后用Dijkstra算法就可以， 但这需要大量的计算。

- 没法处理边权为负值
- Floyd算法(1962)- 赋权有向图，容许出现负权，但不存在权和为负的有向圈。

有兴趣的同学可以找课外资料学习。

(二)、图的代数表示

一个图G=(V，E)由它的顶点与边之间的关联关系唯一确定，也由它的顶点对之间的邻接关系唯一确定。图的这种关系均可以用矩阵来描述，分别称为G的关联矩阵与邻接矩阵。

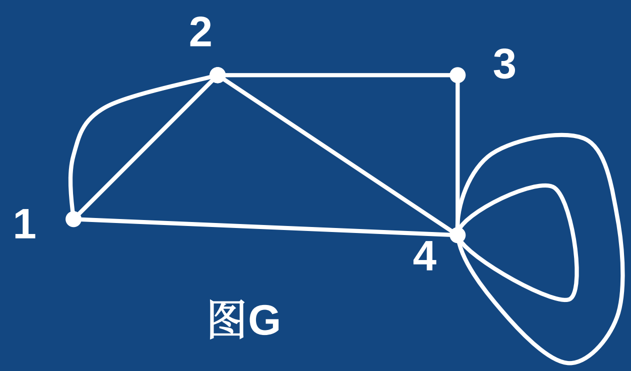
用邻接矩阵或关联矩阵表示图，称为图的代数表示。用矩阵表示图，主要有三个优点：（1）能够把图输入到计算机中；(2）可以用代数方法研究图论；（3）图的矩阵表示在图论应用中具有重要作用。

1、图的邻接矩阵

定义 设G为n 阶图，V={v_1,v_2,…, v_n},

G的邻接矩阵A(G)=$(a_{ij})_{n\times n}$, 其中a_{ij}表示G中顶点v_i与v_j之间的边数。

例如：写出下图G的邻接矩阵：



图G

解：邻接矩阵为：

$$A(G)=\begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

图G的邻接矩阵的性质

(1)非负性与对称性

由邻接矩阵定义知a_{ij}是非负整数，即邻接矩阵是非负整数矩阵；

在图中点v_i与v_j邻接，有v_j与v_i邻接，即$a_{ij} = a_{ji}$.所以，邻接矩阵是对称矩阵。

(2) 同一图的不同形式的邻接矩阵是相似矩阵。

这是因为，同图的两种不同形式矩阵可以通过交换行和相应的列变成一致。

(3) 如果G为简单图，则A(G)为布尔矩阵;行和(列和)等于对应顶点的度数；矩阵元素总和为图的总度数，也就是G的边数的2倍。

（4）G连通的充分必要条件是：A(G)不能通过行与行或者列与列交换变成如下矩阵

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

证明：必要性

反证法： 若不然，设A_{11}对应的顶点是$v_1,v_2,\ldots, v_k$, A_{22}对应的顶点为$v_{k+1},v_{k+2},\ldots, v_n$

显然，v_i (1≤i≤k)与v_j (k+1≤i≤n)不相邻，即G是非连通图。矛盾！

充分性

反证法：假设G不连通， 设G_1与G_2是G的两个不连通的部分，并且设$V(G_1)=\{v_1,v_2,\dots,v_k\}$, $V(G_2)=\{v_{k+1},v_{k+2},\dots,v_n\}$, 在写G的邻接矩阵时，先排$V(G_1)$中点，再排$V(G_2)$中点，则G的邻接矩阵形式必为：

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

矛盾。

（**5**）定理：设 $A^k(G)=(a_{ij}^{(k)})$ ，则**a** $_{ij}^{(k)}$表示顶点**v**$_i$到顶点**v**$_j$的途径**(**道路**)**长度为**k**的途径**(**道路**)**条数。

证明：对**k**作数学归纳法证明。

当**k=1**时，由邻接矩阵的定义，结论成立；
设结论对**k-1**时成立。当为**k**时：

一方面：先计算**A**^{k} **;**

$$A^k=A^{k-1}\cdot A=\left(a_{i1}^{(k-1)}a_{j1}+a_{i2}^{(k-1)}a_{j2}+\cdots+a_{in}^{(k-1)}a_{jn}\right)_{n\times n}$$

另一方面：考虑v_i到v_j的长度为k的途径

设v_m是v_i到v_j的途径中经过的点，且该点和v_j邻接。则v_i到v_j的经过v_m且长度为k的途径数目为：

$$a_{im}^{(k-1)} a_{mj}$$

所以，v_i到v_j的长度为k的途径数目为：

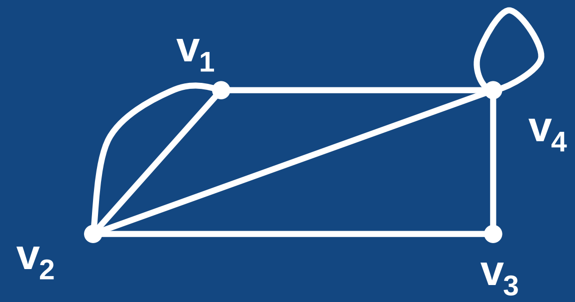
$$a_{i1}^{(k-1)} a_{j1} + a_{i2}^{(k-1)} a_{j2} + \cdots + a_{in}^{(k-1)} a_{jn}$$

即为 $a_{ij}^{(k)}$

问题 1. 定理中 需要加 $i \neq j$ 这个条件么？

问题 2. 定理对所有图成立还是只对简单图成立？

例 求下图中v_1到v_3的途径长度为2和3的条数。



解：由于：

$$A(G)=\begin{pmatrix}0&2&0&1\\2&0&1&1\\0&1&0&1\\1&1&1&1\end{pmatrix}\quad A^2(G)=\begin{pmatrix}5&1&3&3\\1&6&1&4\\3&1&2&2\\3&4&2&4\end{pmatrix}\quad A^3(G)=\begin{pmatrix}5&16&4&12\\16&7&10&12\\4&10&3&8\\12&12&8&13\end{pmatrix}$$

所以，v_1到v_3的途径长度为2和3的条数分别为：3和4。

推论:

(1)若G是简单图，则A^2的元素$a_{ii}^{(2)}$是v_i的度数。

(2) 若G是连通的，对于$i \neq j$, v_i和v_j间距离是使A^n的$a_{ij}^{(n)} \neq 0$的最小整数。

关于（1）如果G是简单图很显然。 如果G是多重图呢？--否　　请说明

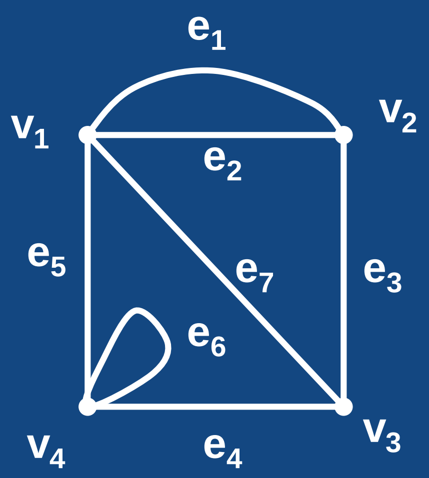
2、图的关联矩阵

(1) 若G是(n, m) 无向图。定义G的关联矩阵：

$$M(G)=(a_{ij})_{n\times m}$$

其中：$a_{ij}=l, v_i$与e_j关联的次数（0, 1, 或2（环））

例如：



$$M(G)=\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 & 0 \end{pmatrix}$$

(2) 关联矩阵的性质

1) 关联矩阵的元素为0，1或2；

2) 关联矩阵的每列和为2；行和为对应顶点度数；

说明: (1) 图的关联矩阵及其性质是网络图论的基础，在电路分析中有重要应用。

(2) 图的关联矩阵比邻接矩阵大得多，不便于计算机存储。但二者都有各自的应用特点。

可达矩阵

定义 设$G=\langle V, E\rangle$为有向图。$V=\{v_1, v_2, \ldots, v_n\}$，令

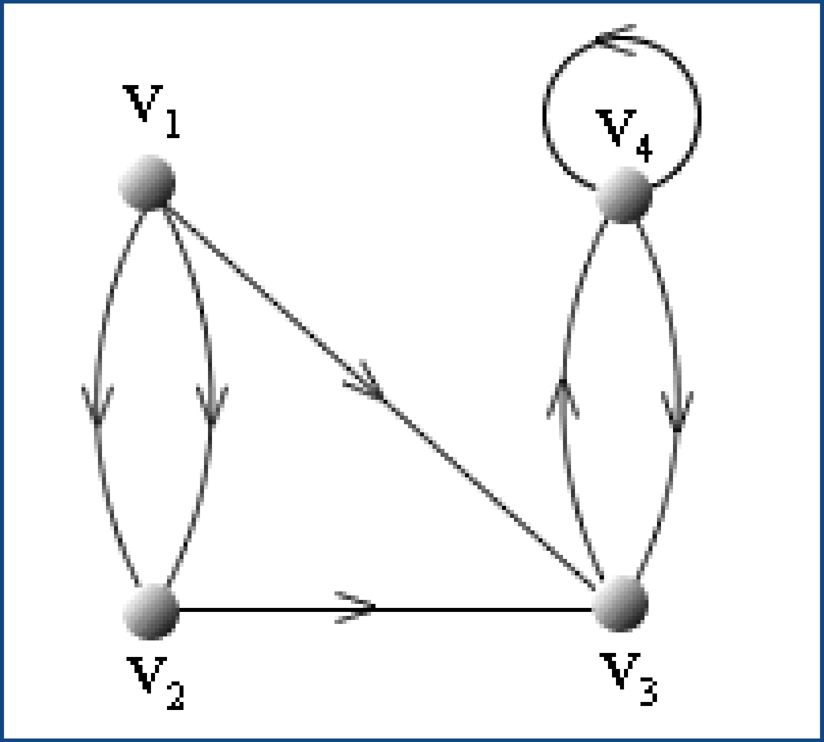
$$r_{ij}=\begin{cases}1 & v_i \text{ 可达 } v_j \\ 0 & \text{否则}\end{cases}$$

称$(r_{ij})_{n\times n}$为G的**可达矩阵**，记作$R(D)$，简记为R。

对有向图， 也可以类似的定义

有向图的邻接矩阵

定义 设有向图$D=\langle V, E\rangle$，$V=\{v_1, v_2, \ldots, v_n\}$，$E=\{e_1, e_2, \ldots, e_m\}$，令$a_{ij}^{(1)}$为顶点$v_i$邻接到顶点$v_j$边的条数，称$(a_{ij}^{(1)})_{n\times n}$为D的**邻接矩阵**，记作$A(D)$，或简记为$A$。



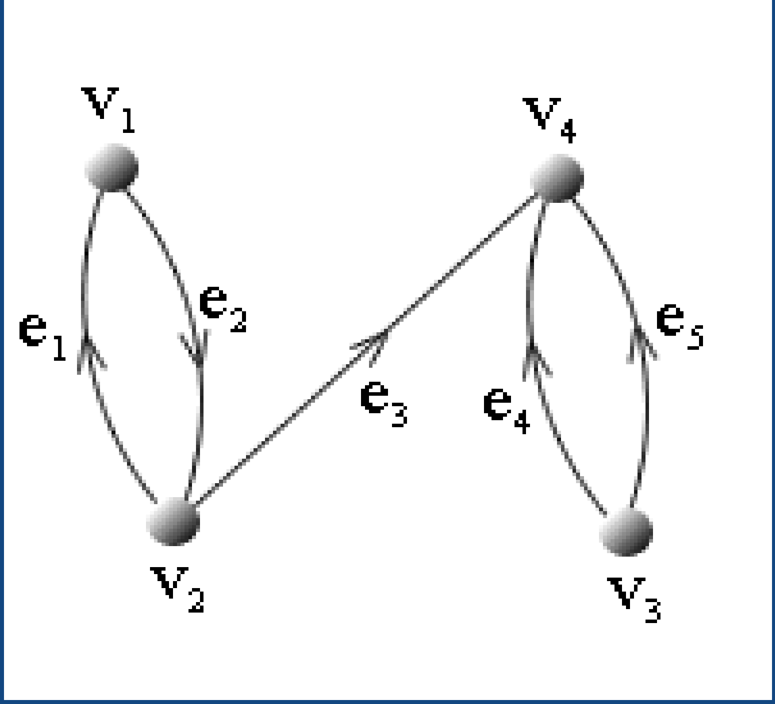
$$A=\begin{bmatrix} 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

有向图的关联矩阵

定义 设有向图$D=\langle V, E\rangle$中无环，$V=\{v_1, v_2, \ldots, v_n\}$，$E=\{e_1, e_2, \ldots, e_m\}$，令

$$m_{ij}=\begin{cases}1 & v_i\text{为}e_j\text{的始点}\\ 0 & v_i\text{为}e_j\text{不关联}\\ -1 & v_i\text{为}e_j\text{的终点}\end{cases}$$

则称$(m_{ij})_{n\times m}$为D的关联矩阵，记作$M(D)$。



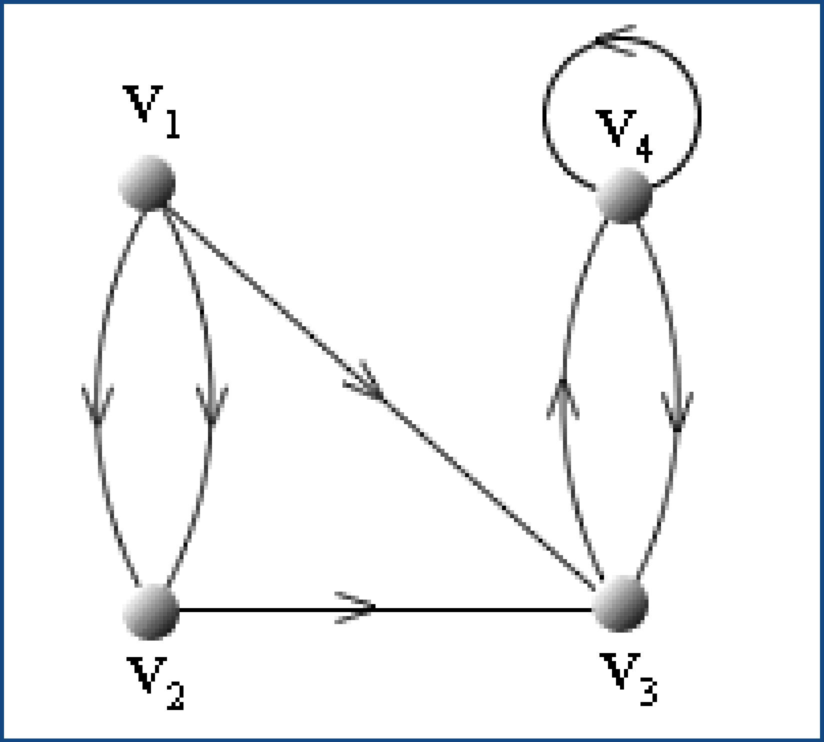
$$M(D)=\begin{bmatrix}-1 & 1 & 0 & 0 & 0\\ 1 & -1 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 1\\ 0 & 0 & -1 & -1 & -1\end{bmatrix}$$

有向图的可达矩阵

定义 设$D=\langle V, E\rangle$为有向图。$V=\{v_1, v_2, \ldots, v_n\}$，令

$$p_{ij}=\begin{cases} 1 & v_i \text{ 可达 } v_j \\ 0 & \text{否则} \end{cases}$$

称$(p_{ij})_{n\times n}$为D的**可达矩阵**，记作$P(D)$，简记为P。



$$P=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$