课时八　图

考点	重要程度	占分	题型
1.图的基本概念	★★★★	2~4	选择、填空、判断
2.图的存储结构	★★★★		
3.图的遍历	★★★★★	4~6	解答、填空

视频讲解更清晰
仅4小时

高数帮

（4）多重图。若图 G 中某两个结点之间的边数多于一条，又允许顶点通过一条边和自己关 联，则图 G 为多重图。

（5）完全图（简单完全图）。对于无向图，任意两个顶点之间都存在边。 对于有向图，任意两个顶点之间都存在方向相反的两条弧。

（6）子图。设有两个图G=（V,E） 和 G’=（V’,E’），若 V’ 是 V 的子集，且 E’ 是 E 的子集，则 称 G’ 是 G 的子图。若有满足 V(G’)= V(G)的子图 G’，则称其为 G 的生成子图。

高数帮

（7）连通、连通图。　在无向图中，若从顶点 v 到顶点 w 有路径存在，则称 v 和 w 是连通的。

若图 G 中任意两个顶点都是连通的，则称图 G 是连通图。

极小连通子图是既要保持图连通又要使得边数最少的子图。

（8）生成树。连通图的生成树是包含图中全部顶点的一个极小连通子图。

（9）顶点的度、入度和出度。

对于无向图，顶点 v 是依附于该顶点的边的条数。无向图的全部顶点的度的和等于边 数的两倍。

对于有向图，入度是以顶点 v 为终点的有向边的数目。出度是以顶点 v 为起点的有向 边的数目。顶点的度等于入度和出度之和。有向图的全部顶点的入度之和与出度之 和相等，并且等于边数。

高数帮

（10）边的权和网。在一个图中，每条边都可以标上具有某种意义的数值，该数值称为该 边的权值。这种边上带有权值的图称为带权图，也称网。

（11）路径、路径长度和回路。

顶点 v_p 到顶点 v_q 之间的一条路径是指顶点序列$v_p,v_{i1},v_{i2},\cdots,v_{im},v_q$ 。

路径上边的数目称为路径长度。第一个顶点和最后一个顶点相同的路径称为回路或环。

题 1.在一个具有n个顶点的无向图中，要连通全部顶点至少需要____条边。

A.n　　B.n+1　　C.n-1　　D.n/2　　答案：C

题 2.在一个图中，所有顶点的度数之和等于边数的（ ）倍。

A.1/2　　B.1　　C.2　　D.4　　答案：C

8.2 图的存储结构

高数帮

图的存储结构有四种：邻接矩阵法、邻接表法、十字链表法、邻接多重表法。

（1）邻接矩阵法

邻接矩阵存储，是指用一个一维数组存储图中顶点的信息，用一个二维数组存储图中 边的信息（即各顶点之间的邻接关系），存储顶点之间邻接关系的二维数组称为邻接矩阵。

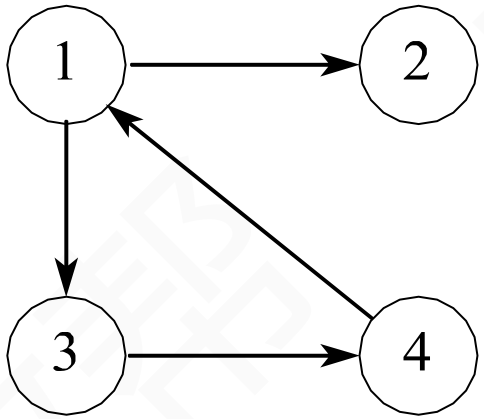
结点数为 n 的图 G(V,E)的邻接矩阵 A 是 n×n 的。将 G 的顶点编号为 $v_1, v_2, \cdots, v_n$。若 $(v_i, v_j) \in E$，则 $A[i][j]=1$，否则 $A[i][j]=0$。

$$A[i][j]=\begin{cases}1 & \text{若}(v_i, v_j)\text{或}\langle v_i, v_j\rangle\text{是}E(G)\text{中的边} \\ 0 & \text{若}(v_i, v_j)\text{或}\langle v_i, v_j\rangle\text{不是}E(G)\text{中的边}\end{cases}$$

8.2 图的存储结构

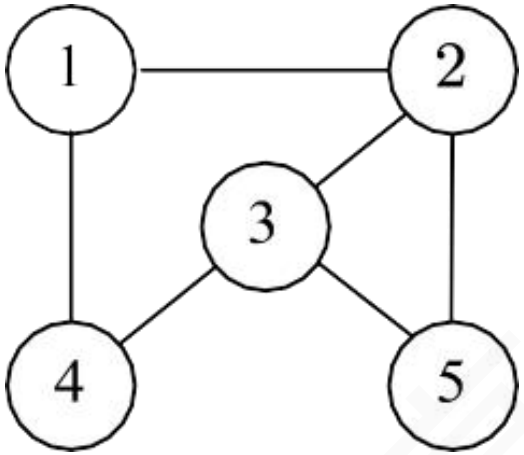
对于带权图而言，若顶点 v_i 和 v_j 之间有边相连，则邻接矩阵中对应项存放着该边对应 的权值，若顶点 v_i 和 v_j 不相连，则用 来表示这两个顶点之间不存在边。

$$A[i][j]=\begin{cases} w_{ij}, & \text{若}(v_i,v_j)\text{或}\langle v_i,v_j\rangle\text{是}E(G)\text{中的边} \\ 0\text{或}\infty, & \text{若}(v_i,v_j)\text{或}\langle v_i,v_j\rangle\text{不是}E(G)\text{中的边} \end{cases}$$

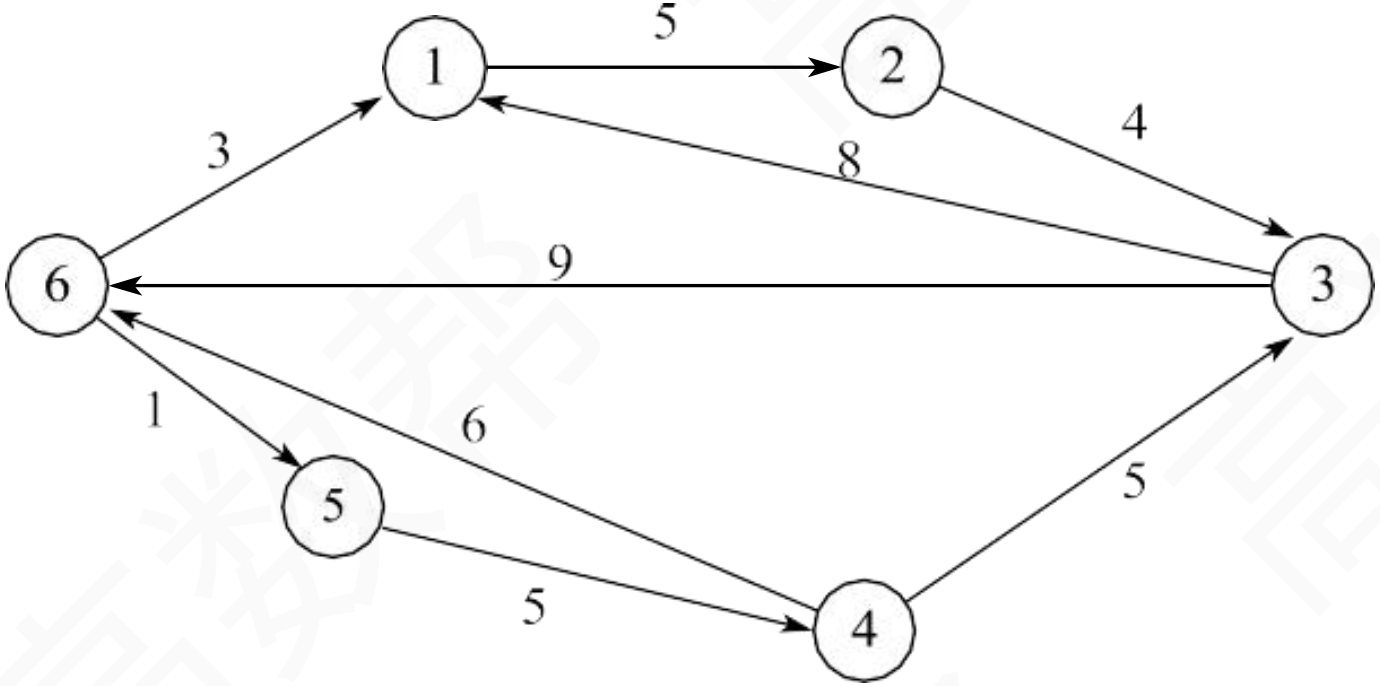


$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

8.2 图的存储结构



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



$$\begin{bmatrix} \infty & 5 & \infty & \infty & \infty & \infty \\ \infty & \infty & 4 & \infty & \infty & \infty \\ 8 & \infty & \infty & \infty & \infty & 9 \\ \infty & \infty & 5 & \infty & \infty & 6 \\ \infty & \infty & \infty & 5 & \infty & \infty \\ 3 & \infty & \infty & \infty & 1 & \infty \end{bmatrix}$$

8.2 图的存储结构

高数帮

图的邻接矩阵存储表示法具有以下特点：

（1）无向图的邻接矩阵一定是个对称矩阵。

（2）对于无向图，邻接矩阵的第 i 行（或第 i 列）非零元素（或非∞元素）的 i 个数正好是第 i 个顶点的度。

（3）对于有向图，邻接矩阵的第 i 行（或第 i 列）非零元素（或非∞元素）的i 个数正好是第 i 个顶点的出度（或入度）。

题 1.在无权图G的邻接矩阵A中，若(vi,vj)或 < vi,vj > 属于图G的边集合，则对应元素A[i][j]等于____，否则等于____。

答案：1;0

高数帮

题 2.已知一个有向图的邻接矩阵表示，计算第i个结点的入度的方法是_____。

答案：求矩阵第i列非零元素之和

题 3.用邻接矩阵存储一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小只与图 中顶点的个数有关，而与图的边数无关。（）

答案：正确

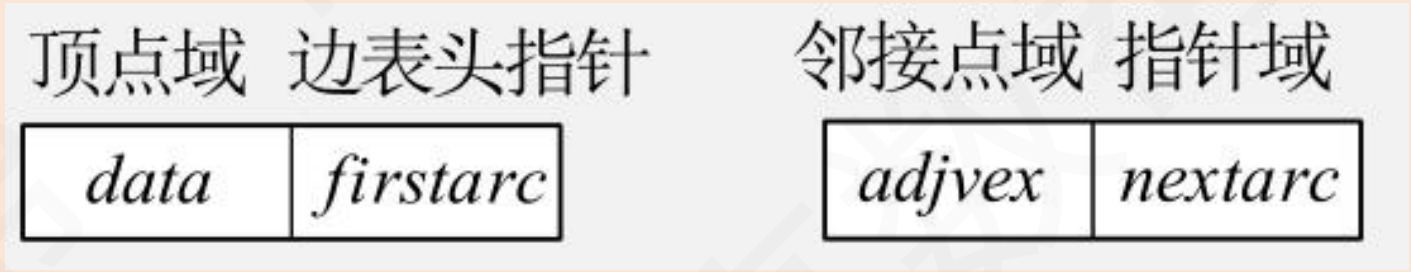
视频讲解更清晰
仅4小时

高数帮

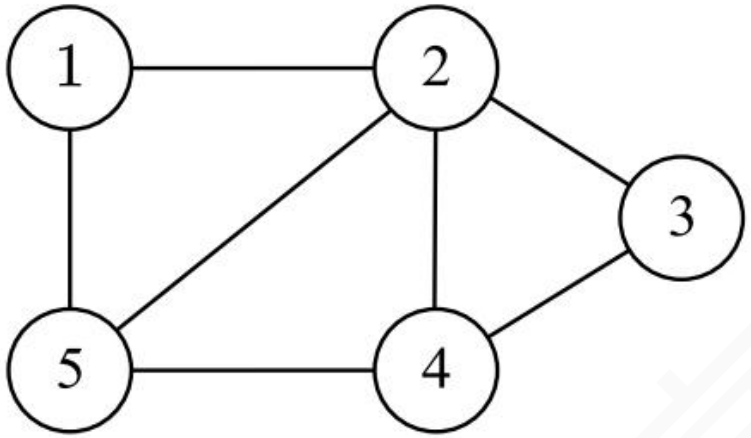
（2）邻接表法

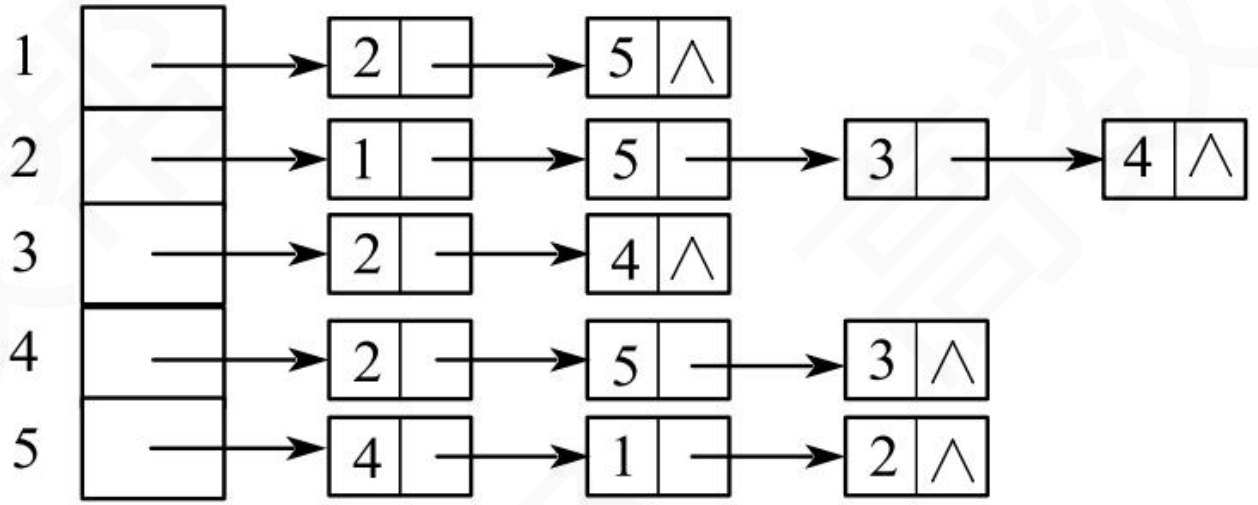
邻接表，是指对图 G 中的每个顶点 v_i 建立一个单链表，第 i 个单链表中的结点表示依附于顶点 v_i 的边，这个单链表就称为顶点 v_i 的边表。

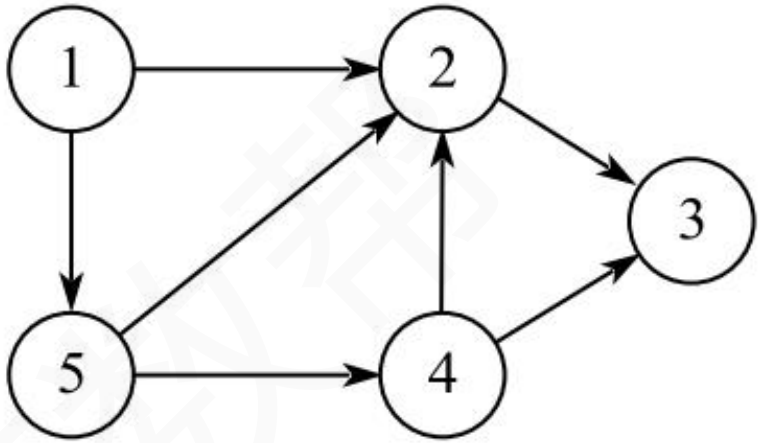
边表的头指针和顶点的数据信息采用顺序存储（称为顶点表），所以在邻接表中存在 两种结点：顶点表结点和边表结点。

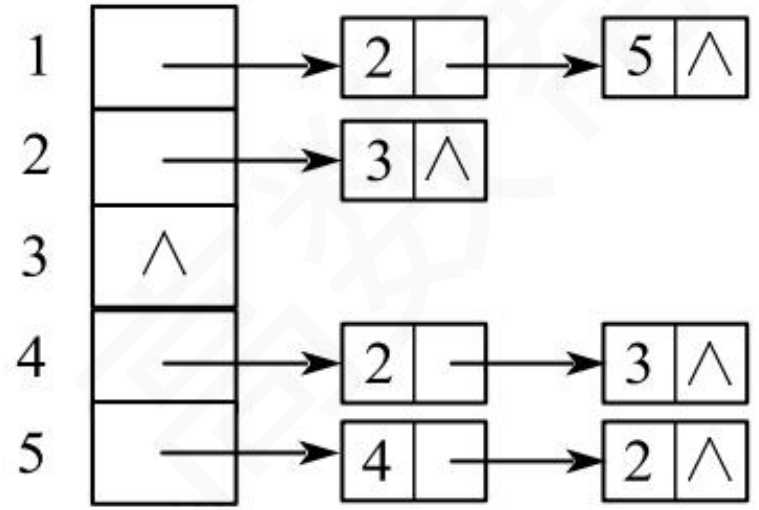


8.2 图的存储结构









8.2 图的存储结构

高数帮

图的邻接表存储表示法具有以下特点：

（1）若 G 为无向图，则所需的存储空间为 $O(|V|+2|E|)$；若 为有向图，则所需的存储 空间为 $O(|V|+|E|)$ 。

（2）对于稀疏图，采用邻接表表示能极大地节省存储空间。

（3）图的邻接表表示不唯一。

（4）在有向图的邻接表表示中，求一个给定顶点的出度只需计算其邻接表中的结点个数。

高数帮

题 4. n个结点， e 条边的无向图邻接表中，有 个头结点和 个表结点。

答案：n,2e

题 5.在有向图的邻接表表示中，下面（ ）最费时间。

A.求某顶点的出度　　B.求某顶点的入度

C.求图中顶点的个数　　D.求从某顶点出发的弧

答案：B

视频讲解更清晰
仅4小时

8.3 图的遍历

高数帮

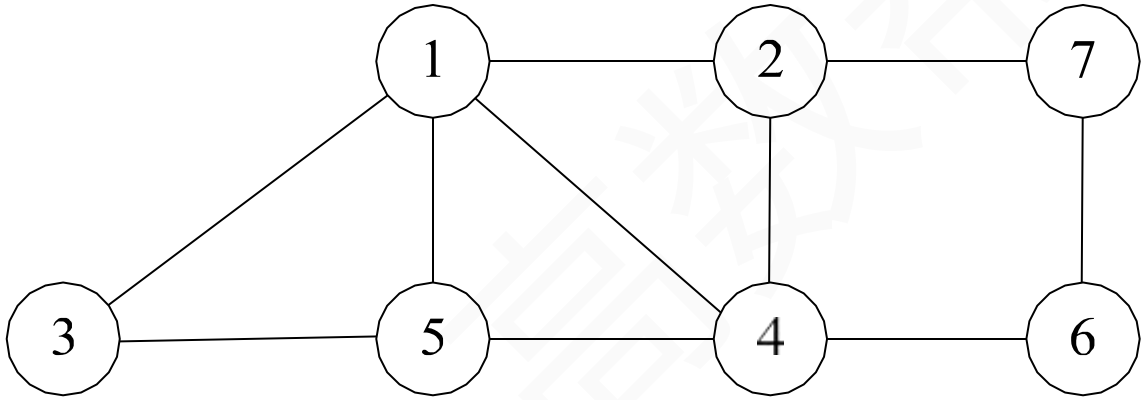
图的遍历是指从图中的某一顶点出发，按照某种搜索方法沿着图中的边对图中的所有顶点访问一次且仅访问一次。

图的遍历算法主要有两种：广度优先搜索和深度优先搜索。

（1） 广度优先搜索

广度优先搜索类似于二叉树的层序遍历算法。其基本思想是：首先访问起始顶点 v，接着由 v 出发，依次访问 v 的各个未访问过的邻接顶点 $w_1,w_2,\cdots,w_i$，然后依次访问 $w_1,w_2,\cdots,w_i$ 的所有未被访问过的邻接顶点；再从这些访问过的顶点出发，访问它们所有 未被访问过的邻接顶点，直至图中的所有顶点都被访问过为止。若此时图中尚有顶点未 被访问，则另选图中的一个未被访问的顶点作为始点，重复上述过程，直至图中所有顶 点都被访问到为止。

题 1. 如下图，从顶点 出发，按照广度优先规则遍历，可能得到的序列是（ ）。



A.1352467　　B.1423756　　C.1234576　　D.1354672

答案：C

高数帮

（2） 深度优先搜索

广度优先搜索类似于树的先序遍历。其基本思想是：首先访问起始顶点 v，然后由 v 出发，访问与 v 邻接且未被访问的任一顶点 w_1，再访问与 w_1 邻接且未被访问的任一顶点w_2……重复上述操作。当不能再继续向下访问时，依次退回到最近被访问的顶点，若它 还有邻接顶点未被访问过，则从该点开始继续上述搜索过程，直至图中所有顶点均被访 问过为止。

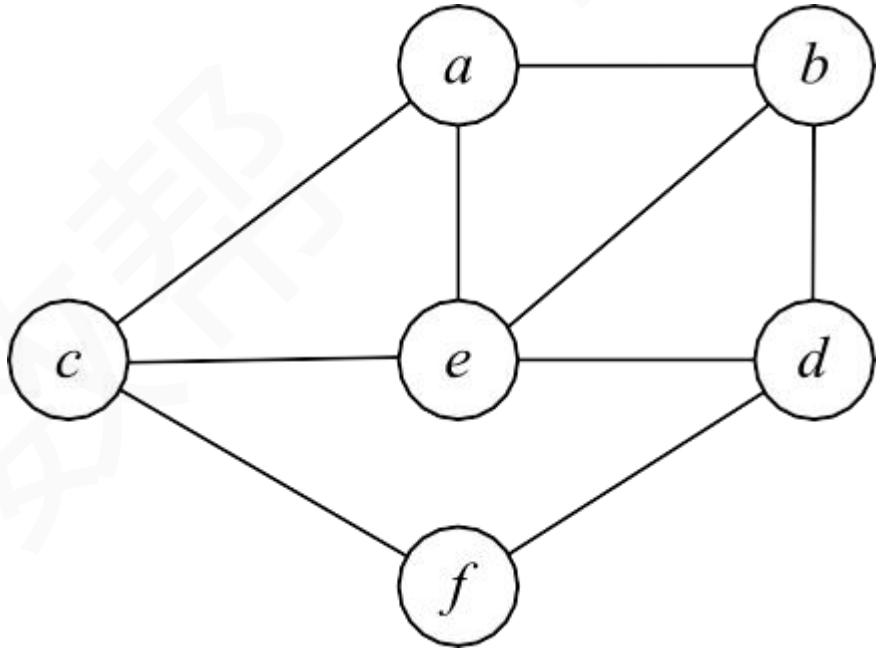
高数帮

题 2. 若无向图 G 中的边的集合 E={(a,b),(a,e),(a,c),(b,e),(e,d),(d,f),(f,c),}，则从顶点 a 出发进行深度优先遍历，可以得到的一种顶点序列为（ ）。

A.aedfcb　　B.acfebd　　C.aebcfd　　D.aedfbc

答案：A

视频讲解更清晰
仅4小时



高数帮

题 3. 深度优先搜索遍历类似于树的_____遍历，广度优先搜索遍历类似于树的_____遍历，它们可以用_____、_____两种数据结构来实现。

答案：先序，层次，栈，队列

题 4. 如果从一个无向图的任意顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一 定是_____。

答案：连通图

课时九　图的应用

考点	重要程度	占分	题型
1.最小生成树	必考	6~10	选择、填空、判断、解答
2.最短路径	★★★★★	4~8	
3.拓扑排序	★★★★★	4~8	
4.关键路径	★★★★★	4~8	

9.1 最小生成树

对一个带权连通无向图，所有生成树中权值之和最小的生成树称为最小生成树。

最小生成树具有如下性质：

（1）最小生成树不一定是唯一的，即最小生成树的树形不唯一。

当带权连通无向图中的各边权值互不相等时，这时最小生成树才是唯一的。

（2）最小生成树的边的权值之和总是唯一的。

（3）最小生成树的边数为顶点数减 1。

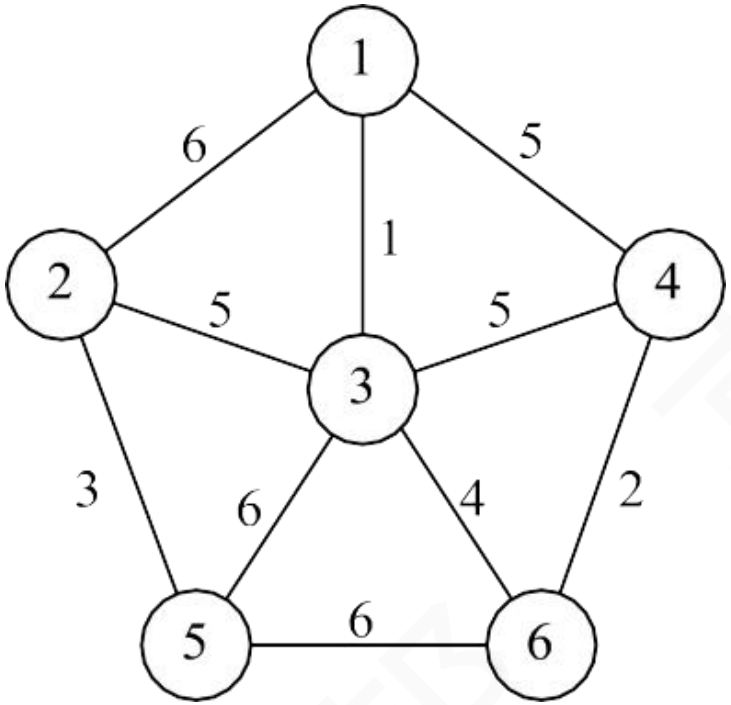
Prim算法：

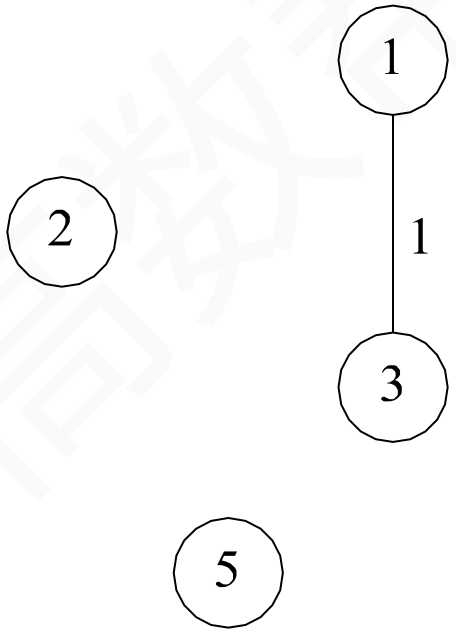
①任取一顶点（或题中已告知顶点），去掉所有边

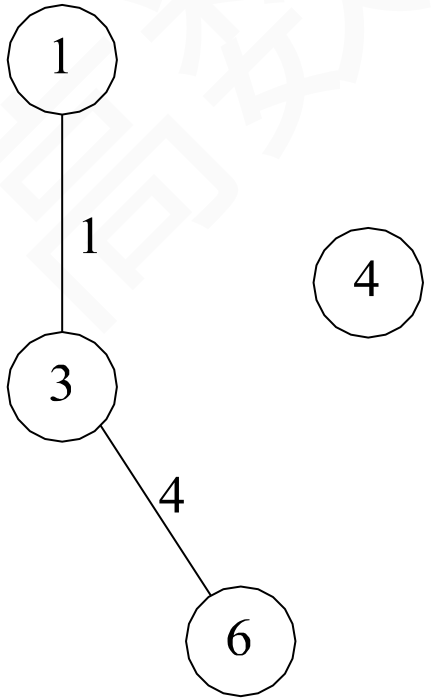
②选择一个与当前顶点集合距离最近的顶点，并将该顶点和相应的边加入进来，同时不形成回路

③重复②，直至图中所有顶点都并入

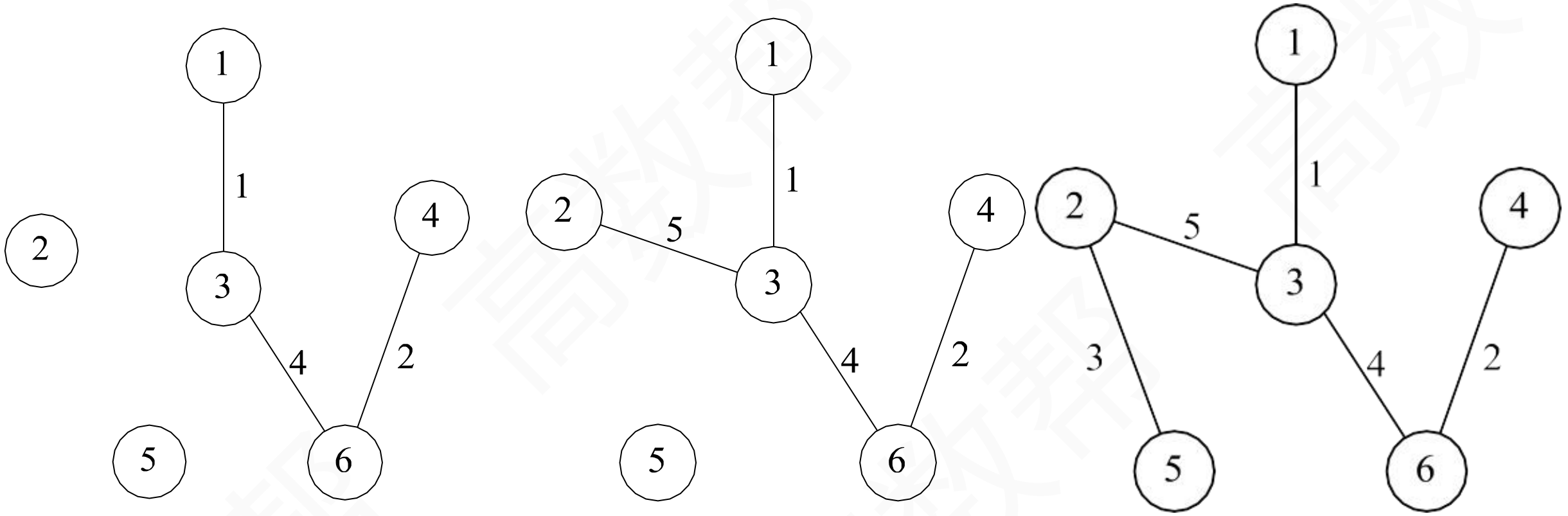
9.1 最小生成树







9.1 最小生成树

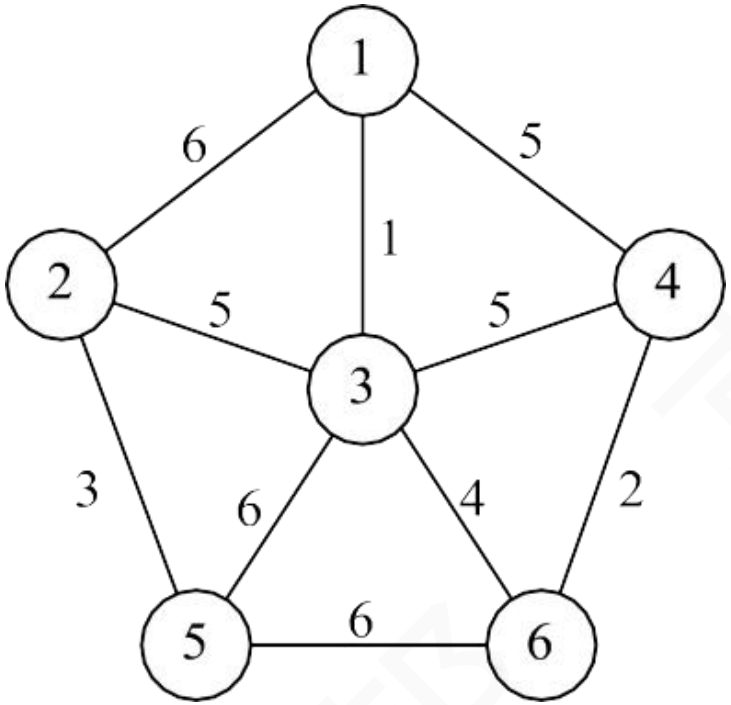


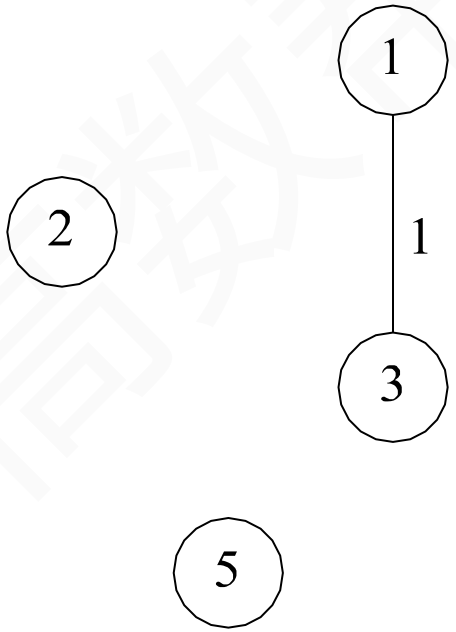
Kruskal 算法：①去掉所有边

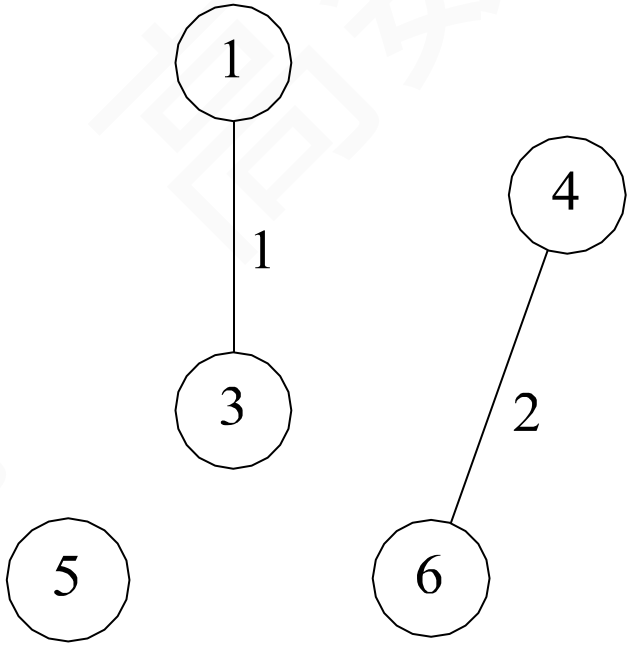
②选边（权最小，且不构成回路）

③一直重复第②步，直至图中所有顶点都并入

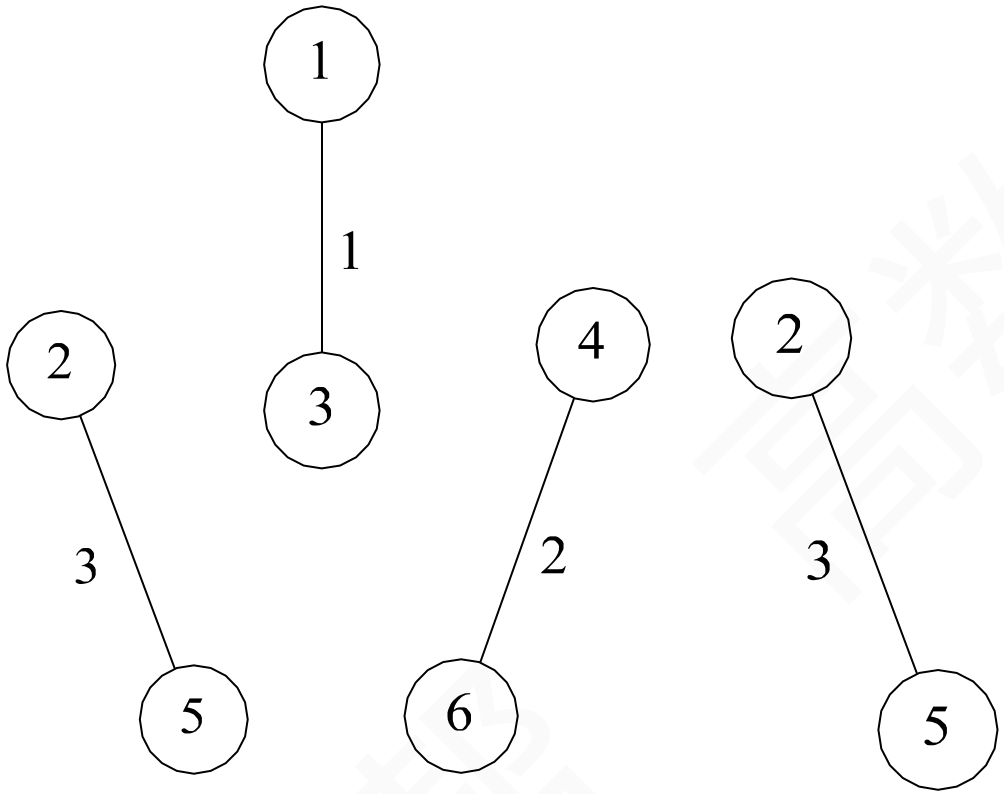
9.1 最小生成树

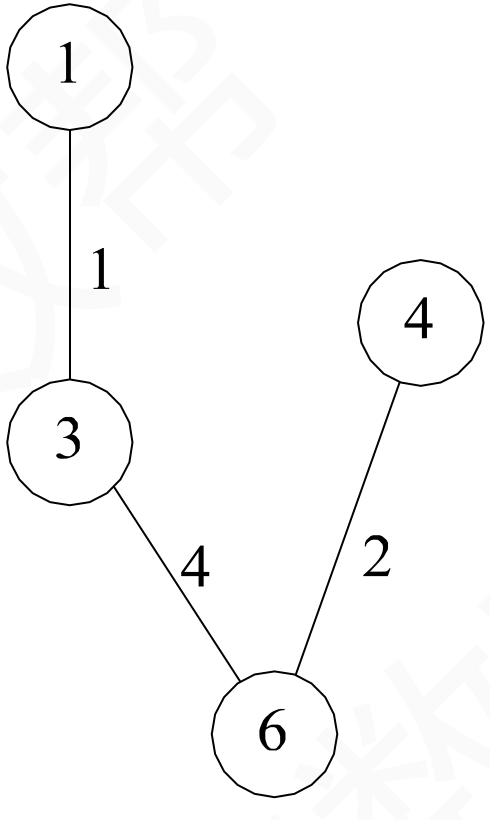


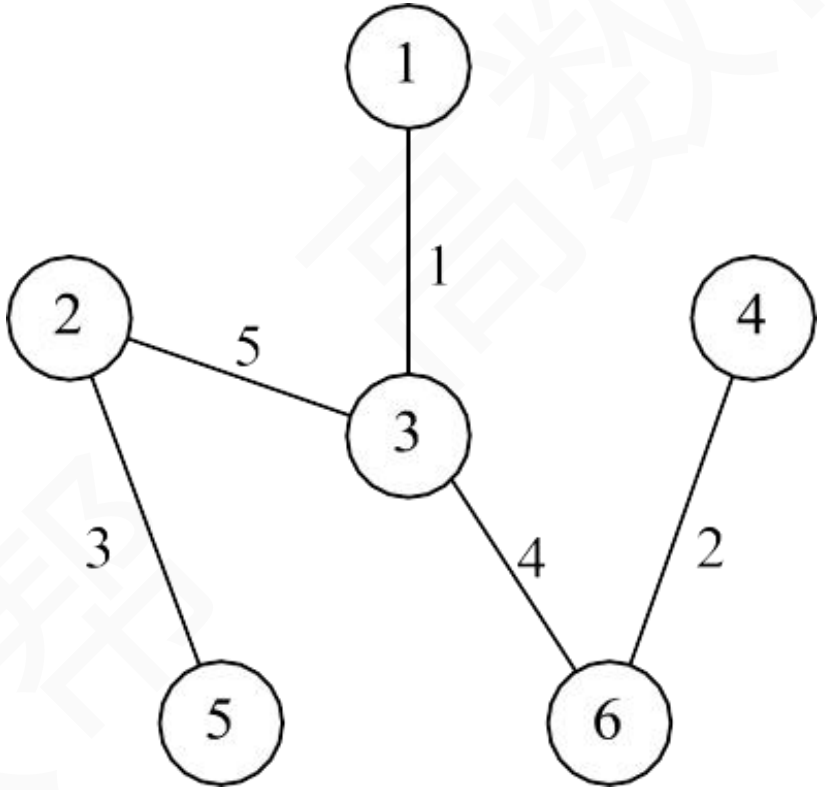




9.1 最小生成树







9.2 最短路径

高数帮

设图$G=<V,E>$（无向图或有向图），给定$W: E\to R$，对于G的每一条边e，称$W(e)$为边e的权，把这样的图称作带权图，记作$G=<V,E,W>$。当$e=(u,v)(<u,v>)$时，把$W(e)$记作$W(u,v)$。

设P是G中的一条通路，P中所有边的权之和称作P的长度，记作$W(P)$，即$W(P)=\sum_{e\in E(p)} W(e)$。类似地，可以定义为回路$C$的长度$W(C)$。

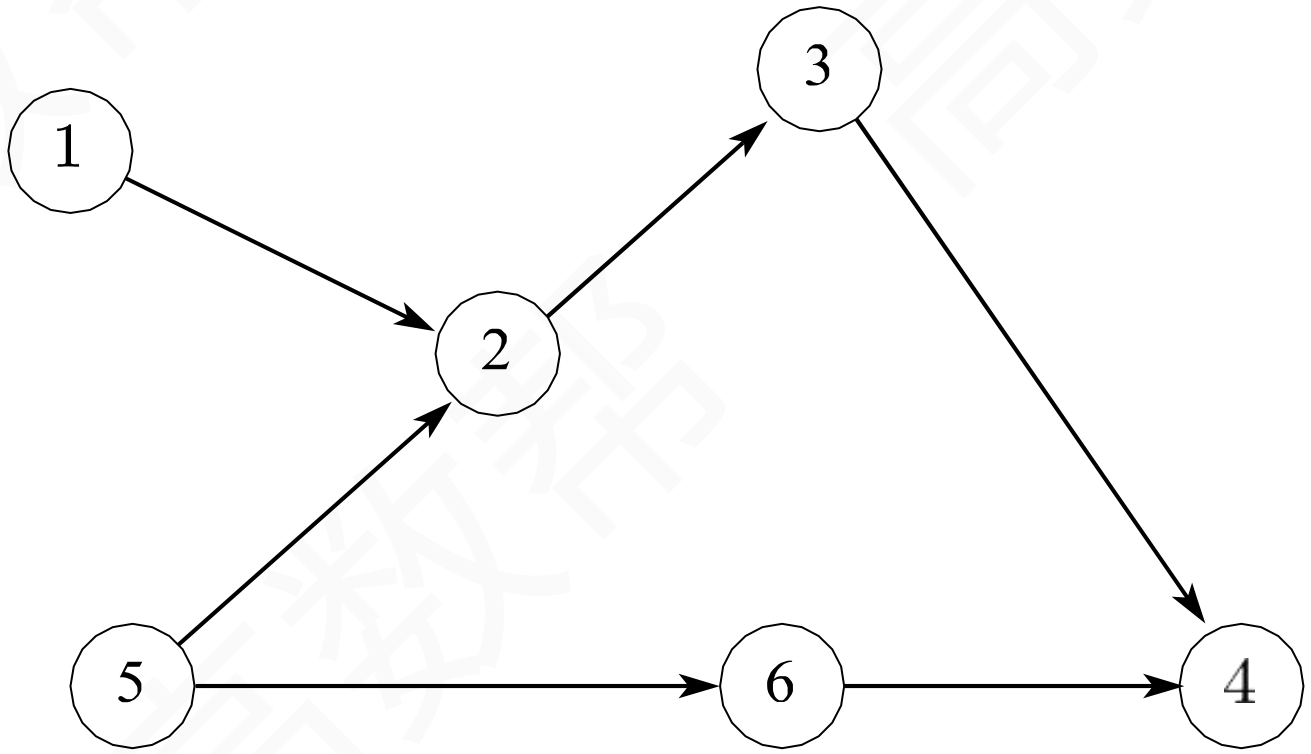
设带权图$G=<V,E,W>$（无向图和有向图），其中每一条边e的权$W(e)$为非负实数。$\forall u,v\in V$，当u和v连通（u可达v）时，称从u到v的最短路径，称其长度为从u到v的距离，记作$d(u,v)$。约定：$d(u,v)=0$；当u和v不连通（u不可达v）时，$d(u,v)=+\infty$。

最短路径问题：给定带权图$G=<V,E,W>$及顶点u和v，其中每一条边e的权$W(e)$为非负实数，求从u到v的最短路径。

9.2 最短路径

题 1. 带权图 G 如图所示，求从v_1到其余各点的最短路径和距离。

答案：152364



步骤 / 顶点	1	2	3	4	5	6	7
v_1	0						
v_2	∞	3					
v_3	∞	7	5				
v_4	∞	5	5	5			
v_5	∞	∞	9	8	8	8	
v_6	∞	∞	∞	∞	7		
v_7	∞	∞	∞	∞	13	9	9
	0	3	5	5	7	8	9
	v_1	v_2	v_3	v_4	v_5	v_6	v_7

高数帮

从v_1到其余各点的最短路径和距离如下：

v_1v_2 $\quad d(v_1,v_2)=3$

$v_1v_2v_3$ $\quad d(v_1,v_3)=$

v_1v_4 $\quad d(v_1,v_4)=$

$v_1v_2v_3v_5$ $\quad d(v_1,v_5)=$

$v_1v_4v_6$ $\quad d(v_1,v_6)=$

$v_1v_4v_6v_7$ $\quad d(v_1,v_7)=$

视频讲解更清晰
仅4小时

9.3 拓扑排序

高数帮

AOV 网：若用 DAG 图表示一个工程，其顶点表示活动，用有向边 $<V_i,V_j>$表示活动 V_i 必须先于活动 V_j 进行的这样一种关系，则将这种有向图称为顶点表示活动的网络，记为 AOV 网。

在 AOV 网中，活动 V_i 是活动 V_j 的直接前驱，活动 V_j 是活动 V_i 的直接后继，这种前驱和后 继关系具有传递性，且任何活动 不能以它作为自己的前驱或后续。

9.3 拓扑排序

高数帮

在图论中，由一个有向无环图的顶点组成的序列，当且仅当满足下列条件时，称为该图的 一个拓扑排序：

①每个顶点出现且只出现一次。

②若存在一条从顶点 A 到顶点 B 的路径，则在排序中顶点 B 出现在顶点 A 的后面。 每个 AOV 网都有一个或多个拓扑排序序列。

9.3 拓扑排序

高数帮

拓扑排序的算法的步骤：

①从 AOV 网中选择一个没有前驱的顶点并输出。

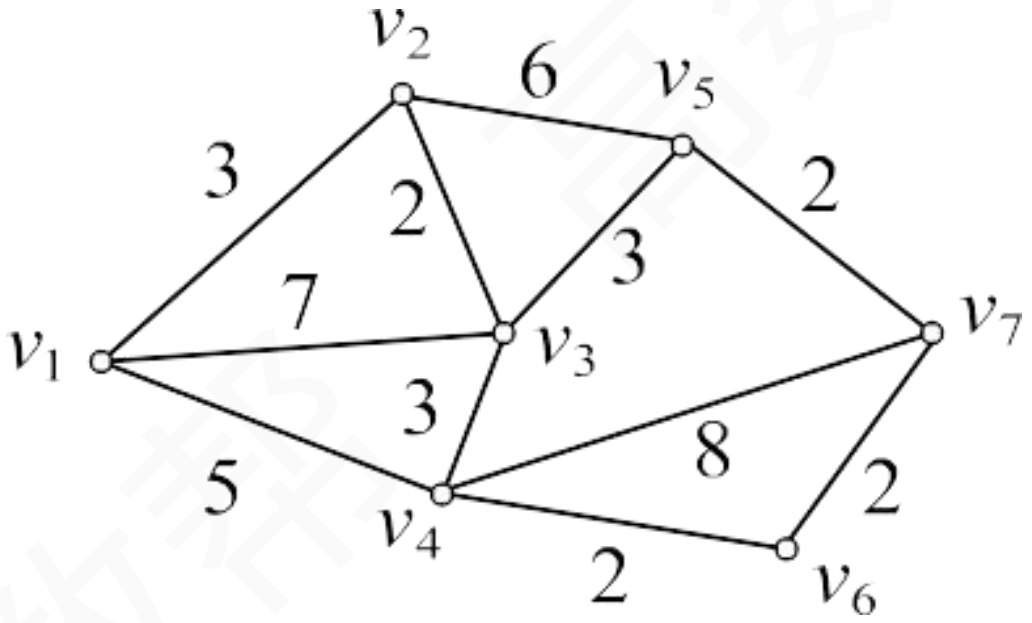
②从网中删除该顶点和所有以它为起点的有向边。

③重复①和②直到当前的 AOV 网为空或当前网中不存在无前驱的顶点为止。后一种情况 说明有向图中必然存在环。

9.3 拓扑排序

题 1. 下图的一个拓序排序序列为_____。

视频讲解更清晰
仅4小时



9.4 关键路径

高数帮

在带权有向图中，以顶点表示事件，以有向边表示活动，以边上的权值表示完成该活 动的开销（如完成活动所需的时间），称之为用边表示活动的网络，简称 AOE网。 AOE 网 和 AOV 网都是有向无环图，不同之处在于它们的边和顶点所代表的含义是不同的，AOE 网中的边有权值；而 AOV 网中的边无权值，仅表示顶点之间的前后关系。

从开始顶点到结束顶点的所有路径中，具有最大路径长度的路径称为关键路径。而把 关键路径上的活动称为关键活动。

题 1. 关键路径是事件网络中（ ）。

A.从源点到汇点的最短路径　B.从源点到汇点的最长路径

C.最长的回路　　D.最短的回路　　答案：B

9.4 关键路径

寻找关键活动时所用到的几个参量：

（1）事件 的最早发生时间： 指从源点 到顶点 的最长路径长度。

可用下面的递推公式来计算：

$ve(k)$(源点) =0

$ve(k) = Max\{ve(j) + Weight(v_j, v_k)\}$， v_k为v_j的任意后续，$Weight(v_j, v_k)$表示$\langle v_j, v_k \rangle$上的权值。

注意：计算$vl(k)$值时，按从前往后的顺序进行。

（2）事件 v_k 的最迟发生时间 $vl(k)$ ： 指在不推迟整个工程完成的前提下，即保证它的后续 事件 v_j 在其最迟发生时间 vl(j) 能够发生时，该事件最 迟必须发生的时间。

高数帮

vl(汇点)=ve (汇点)，

$vl(k)=Min\{vl(j)-Weight(v_k,v_j)\}$ ，v_k 为 v_j的任意前驱

注意：在计算$vl(k)$时，按从后往前的顺序进行。

（3）活动 a_i 的最早开始时间e(i)：指该活动弧的起点所表示的事件的最早发生时间。若边$\langle v_k,v_j\rangle$ 表示活动a_i，则有$e(i)=ve(k)$ 。

（4）活动 a_i 的最早开始时间l(i)：指该活动弧的起点所表示的事件的最早发生时间。若边$\langle v_k,v_j\rangle$表示活动 a_i ，则有$l(i)=vl(j)-Weight(v_k,v_j)$ 。

（5）一个活动 a_i 的最迟开始时间 l(i) 和其最早开始时间 e(i)的差额 $d(i)=l(i)-e(i)$

$l(i)-e(i)=0$ 即 $l(i)=e(i)$ 的活动 a_i 是关键活动。

9.4 关键路径

求关键路径的算法步骤如下：

①从源点出发，令 ve (源点)= 0，按拓扑有序求其余顶点的最早发生时间ve ()。

②从汇点出发，令 vl (汇点)= ve(汇点)，按逆拓扑有序求其余顶点的最迟发生时间 vl ()。

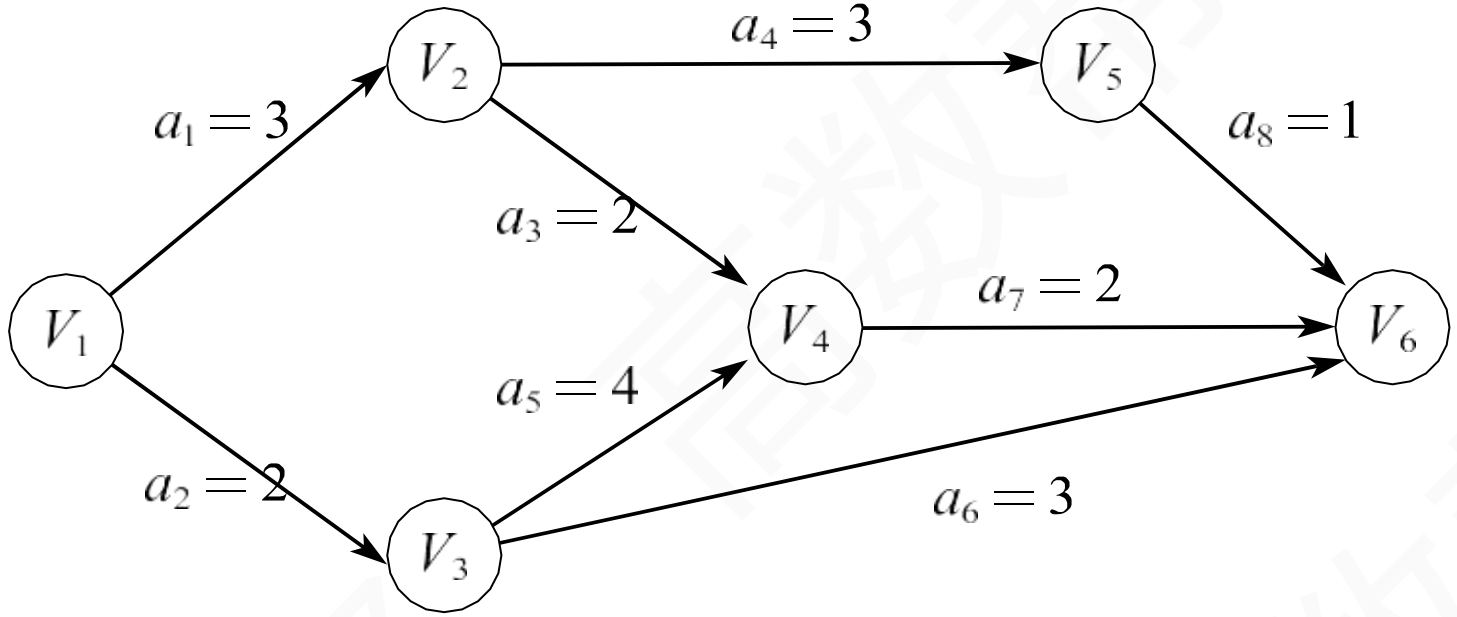
③根据各顶点的 ve ()值求所有弧的最早开始时间 e ()。

④根据各顶点的 vl ()值求所有弧的最迟开始时间 l ()。

⑤求 AOE 网中所有活动的差额 d ()，找出所有 d()= 0的活动构成关键路径。

9.4 关键路径

题 2.求下图的关键路径。



	v_1	v_2	v_3	v_4	v_5	v_6
$ve(i)$	0	3	2	6	6	8
$vl(i)$	0	4	2	6	7	8

9.4 关键路径

高数帮

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
$e(i)$	0	0	3	3	2	2	6	6
$l(i)$	1	0	4	4	2	5	6	7
$l(i)-e(i)$	1	0	1	1	0	3	0	1

得到关键路径 (v_1,v_2,v_4,v_6)。

注意：（1）关键路径上的所有活动都是关键活动，因此可以加快关键活动来缩短整个工程的工期。

（2）网中的关键路径并不唯一，且对于有几条关键路径的网，只提高一条关键路径上的关 键活动并不能缩短整个工程的工期，只有加快那些包括在所有关键路径上的关键活动 才能达到缩短工期的目的。