图论应用

一、最小生成树

1.1 最小生成树的概念

由生成树的定义可知，无向连通图的生成树不是唯一的。连通图的一次遍历所经过的边的集合及图中所有顶点的集合就构成了该图的一棵生成树，对连通图的不同遍历，就可能得到不同的生成树。可以证明，对于有 n 个顶点的无向连通图，无论其生成树的形态如何，所有生成树中都有且仅有 n−1 条边。

如果无向连通图是一个网，那么，它的所有生成树中必有一棵边的权值总和最小的生成树，我们称这棵生成树为最小生成树，简称为最小生成树。

1.2 普里姆(Prim)算法

假设 G=（V，E）为一网图，

设置两个新的集合 U 和 T，**U存最小生成树的顶点，T存最小生成树的边**

令集合 U 的初值为 U={u1}（假设构造最小生成树时，从顶点 u1 出发），集合 T 的初值为 T={}。

Prim 算法的思想是:从所有 u∈U，v∈V−U 的边中，选取具有最小权值的边（u，v），将点 v 加入集合 U 中，将边（u，v）加入集合 T 中，如此不断重复，直到 U=V 时，最小生树构造完毕，这时集合 T 中包含了最小生成树的所有边。

也就是一直选与顶点相连的最小的边，然后到下一节点，选择该节点相连的最小的边，前提不能出现回路即可

Prim 算法可用下述过程描述，其中用 w_{uv} 表示顶点 u 与顶点 v 边上的权值。

（1）U={u1},T={};

（2）while (U≠V)do

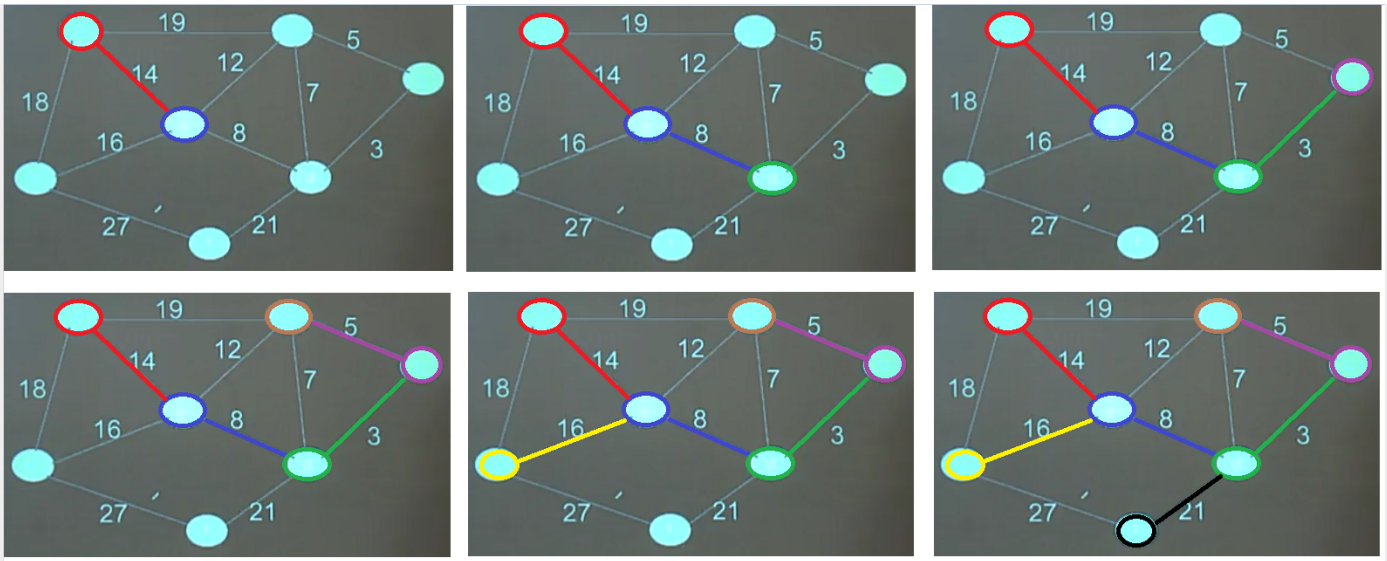
(u，v)=min{w_{uv}；u∈U，v∈V−U }

T=T+{(u，v)}

U=U+{v}

（3）结束。

Prim 算法的时间复杂度为 $O(n^2)$，与网中的边数无关，因此适用于求边稠密的网的最小生成树。



14+8+3+5+16+21=67

1.3 卡鲁斯卡尔(Kruskal)算法---加边法

Kruskal 算法是一种按照网中边的权值递增的顺序构造最小生成树的方法。其基本思想是：设无向连通网为 G=（V，E），令 G 的最小生成树为 T，其初态为 T=（V，{}），

初始时，只有顶点，没有边

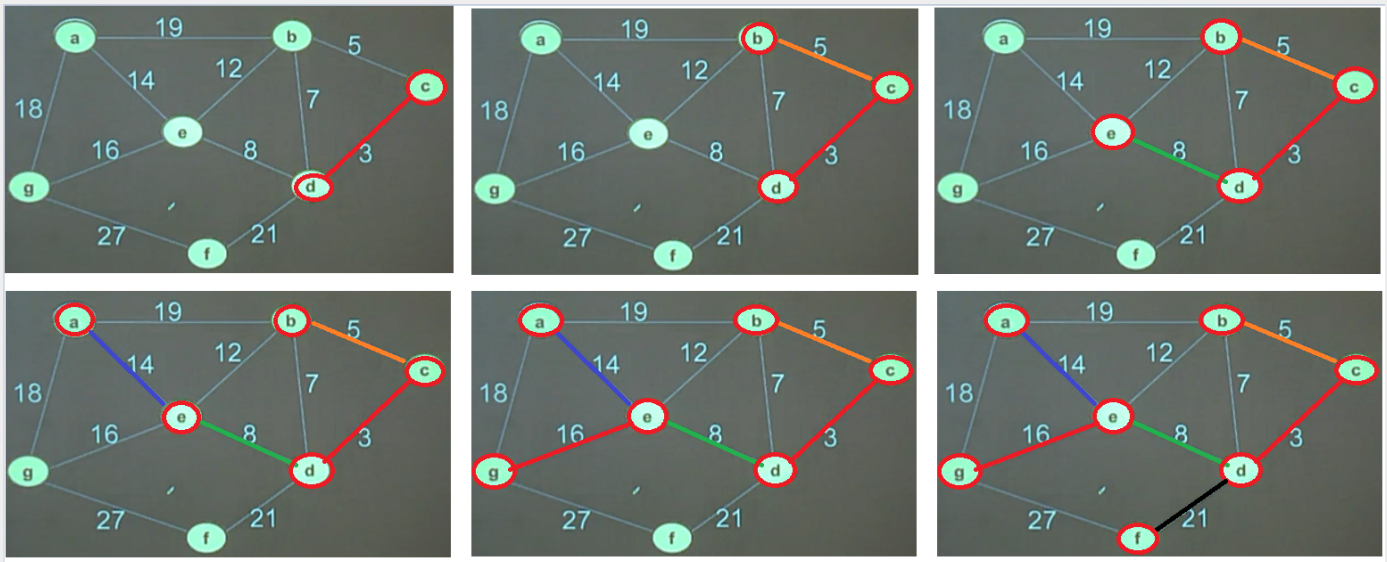
将边按照权值排序

按照权值加边

在不构成回路的情况下

使得连通分量只有一个，即构成一棵最小生成树

Kruskal 算法需对 e 条边按权值进行排序，时间复杂度为 O(eloge)(e 为网中边的数目) 因此适用于求边稀疏的网的最小生成树。



3+5+8+14+16+21=67

二、拓扑排序

2.1 AOV(Activity On Vertex Network)网

所有的工程或者某种流程可以分为若干个小的工程或阶段，这些小的工程或阶段就称为活动。若以图中的顶点来表示活动，有向边表示活动之间的优先关系，则这样活动在顶点上的有向图称为 AOV 网。在 AOV 网中，若从顶点 i 到顶点 j 之间存在一条有向路径，称顶点 i 是顶点 j 的前驱，或者称顶点 j 是顶点 i 的后继。若<i,j>是图中的弧，则称顶点 i 是顶点 j 的直接前驱，顶点 j 是顶点 i 的直接后驱。 **仅仅是考虑优先关系而已，不考虑权值**

2. 拓扑排序

AOV 网所代表的一项工程中活动的集合显然是一个偏序集合。为了保证该项工程得以顺利完成，必须保证 AOV 网中不出现回路；否则，意味着某项活动应以自身作为能否开展的先决条件，这是荒谬的。

测试 AOV 网是否具有回路（即是否是一个有向无环图）的方法，就是在 AOV 网的偏序集合下构造一个线性序列，该线性序列具有以下性质：**有序的保持顺序**

① 在 AOV 网中，若顶点 i 优先于顶点 j ，则在线性序列中顶点 i 仍然优先于顶点 j；

② 对于网中原来没有优先关系的顶点 i 与顶点 j，在线性序列中也建立一个先后关系，或者顶点 i 优先于顶点 j ，或者顶点 j 优先于 i。**无序的人为定义顺序**

满足这样性质的线性序列称为拓扑有序序列，构造拓扑序列的过程称为拓扑排序。也可以说拓扑排序就是由某个集合上的一个偏序得到该集合上的一个全序的操作。

若某个 AOV 网中所有顶点都在它的拓扑序列中，则说明该 AOV 网不会存在回路，这时的拓扑序列集合是 AOV 网中所有活动的一个全序集合。显然，对于任何一项工程中各个活动的安排，必须按拓扑有序序列中的顺序进行才是可行的。

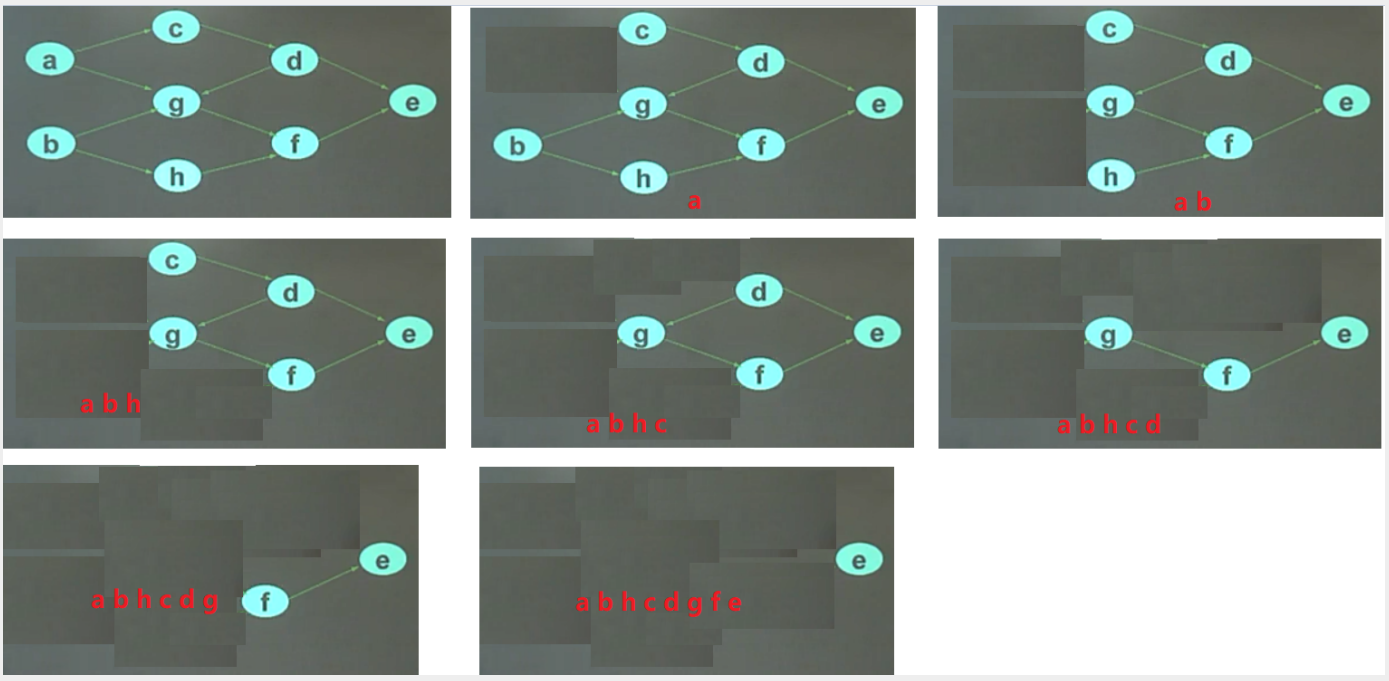
3.拓扑排序算法

对 AOV 网进行拓扑排序的方法和步骤是：

①从 AOV 网中选择一个没有前驱的顶点（该顶点的入度为 0）并且输出它

②从网中删去该顶点，并且删去从该顶点发出的全部有向边；

③重复上述两步，直到剩余的网中不再存在没有前驱的顶点为止。



三、关键路径

3.1 AOE网

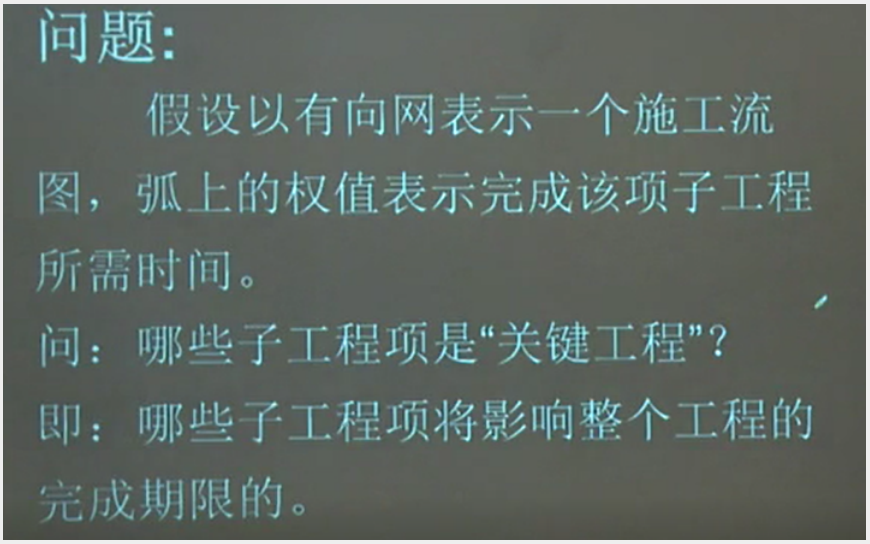
若在带权的有向图中，以**顶点表示事件**，以**有向边表示活动**，**边上的权值表示活动的开销（如该活动持续的时间）**，则此带权的有向图称为AOE网。

AOE 网具有以下两个性质：

①只有在某顶点所代表的事件发生后，从该顶点出发的各有向边所代表的活动才能开始。

② 只有在进入一某顶点的各有向边所代表的活动都已经结束，该顶点所代表的事件才能发生。

3.2 关键路径与关键活动



关键路径

由于 AOE 网中的某些活动能够同时进行，故完成整个工程所必须花费的时间应该为源点到终点的最大路径长度（这里的路径长度是指该路径上的各个活动所需时间之和）。具有最大路径长度的路径称为关键路径。

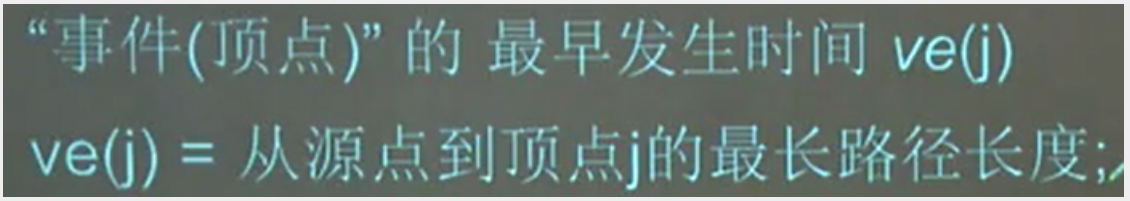
关键活动: **该活动的最早开始时间=活动的最晚开始时间**

3.3 事件的最早发生时间

ve[k]是指从源点到顶点的最大路径长度代表的时间，这个时间决定了所有从顶点发出的有向边所代表的活动能够开工的最早时间。根据 AOE 网的性质，只有进入 vk 的所有活动<vj,vk>都结束时，vk 代表的事件才能发生；而活动< vj, vk>的最早结束时间为 ve[j]+dut(< vj, vk>)。所以计算 vk 发生的最早时间的方法如下：**工期长，所以事件早开始，故为事件的最早发生时间**

$$\begin{cases} ve[1]=0 \\ ve[k]=Max\{ve[j]+dut(<vj, vk>)\} & <vj, vk>\in p[k] \end{cases} \quad (式\ 1\text{-}4\text{-}1)$$

其中，p[k]表示所有到达 vk 的有向边的集合；dut(< vj, vk>)为有向边< vj,vk>上的权值。

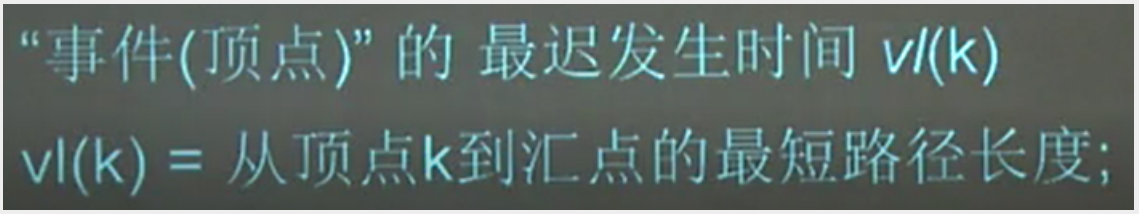


3.4 事件的最晚发生时间

vl[k]是指在不推迟整个工期的前提下，事件 vk 允许的最晚发生时间。设有向边< vk，vj>代表从 vk 出发的活动，为了不拖延整个工期,vk 发生的最迟时间必须保证不推迟从事件 vk 出发的所有活动< vk，vj>的终点 vj 的最迟时间 vl[j]。vl[k] 的计算方法如下：

$$\begin{cases} vl[n]=ve[n] \\ vl[k]=Min\{vl[j]-dut(<vk，vj>)\} & <vk, vj>\in s[k] \end{cases}$$

其中，s[k]为所有从 vk 发出的有向边的集合。



3.5 活动发生的最早和最晚发生时间

（3）活动 ai 的最早开始时间 e[i]

若活动 ai 是由弧<vk,vj>表示，根据 AOE 网的性质，只有事件 vk 发生了，活动 ai 才能开始。也就是说，活动 ai 的最早开始时间应等于事件 vk 的最早发生时间。因此，有：

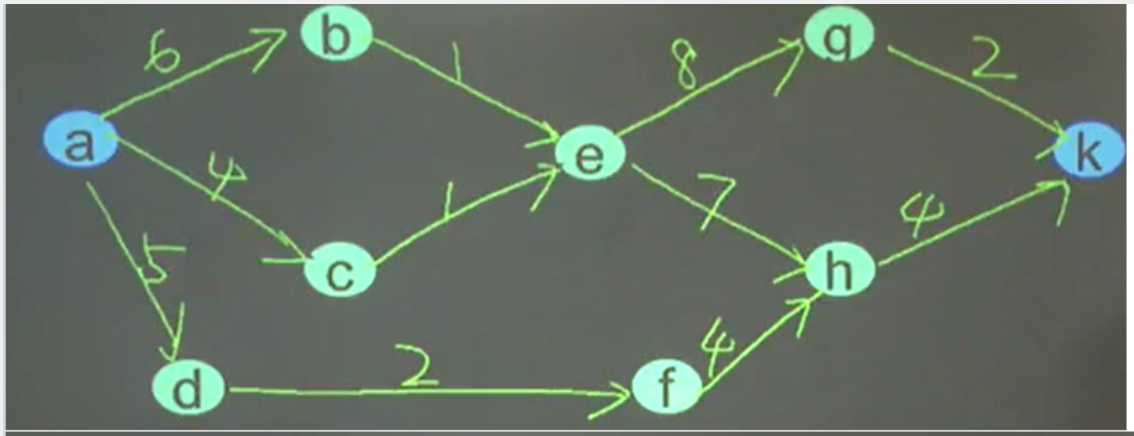
$$e[i]=ve[k]$$

（4）活动 ai 的最晚开始时间 l[i]

活动 ai 的最晚开始时间指，在不推迟整个工程完成日期的前提下，必须开始的最晚时间。若 由弧< vk,vj>>表示，则 ai 的最晚开始时间要保证事件 vj 的最迟发生时间不拖后。因此，应该有：

$$l[j]=vl[j]-dut(<vk,vj>)$$

根据每个活动的最早开始时间 e[i]和最晚开始时间 l[i]就可判定该活动是否为关键活动也就是那些 l[i]=e[i]的活动就是关键活动，而那些 l[i]>e[i]的活动则不是关键活动，l[i]-e[i]的值为活动的时间余量。关键活动确定之后，关键活动所在的路径就是关键路径。



	a	b	c	d	e	f	g	h	k
ve	0	6	4	5	7	7	15	14	18
vl	0	6	6	8	7	10	16	14	18

	ab	ac	ad	be	ce	df	eg	eh	fh	gk	hk
权	6	4	5	1	1	2	8	7	4	2	4
e	0	0	0	6	4	5	7	7	7	15	14
l	0	2	3	6	6	8	8	7	10	16	14

四、最短路径