离散数学II
—道路与回路的经典问题2

周旻
清华大学软件学院
软件工程与系统研究所

2024年5月11日
Saturday

第二章 道路与回路

- 道路与回路的定义和相关概念
- 道路与回路的判定方法
- 欧拉道路与回路
- 哈密顿道路与回路
- 旅行商问题
- 最短路径
- 关键路径
- 中国邮路

关键路径

关键路径

- 问题的提出
  - 一项工程都要由很多工序组成
    - 每个工序的执行时长是可预知的；
    - 这些工序相互约束，只有在某些工序完成之后，一个新的工序才能开始；
    - 一般情况下这种关系是预知的，而且也能预计完成每个工序所需要的时间；
  - 人们往往需要知道
    - 完成整个工程任务最少需要多少时间？
    - 影响工程进度的关键工序是哪几个？

关键路径

- 关键路径问题
  - 关键路径：至少需要多少时间才能完成
  - 最早启动/完工时间
  - 结点最晚完工时间
  - 工序最晚启动时间
  - 工序最大允许延误时间

关键路径

- 图论模型
  - **PT图(Potential task graph)：**
    ＋结点表示工序
    ＋有向边(i, j)表示工序i完成之后工序j才能启动
    ＋边权w_i表示工序i所需的时间
  - **PERT图(Programme evaluation and review technique)**
    ＋结点为工序之间的关系
    ＋v_k是e_i的终点、e_j的始点表示工序e_i完成后e_j才能开始
    ＋有向边表示工序
    ＋边权w_i表示该工序所需时间

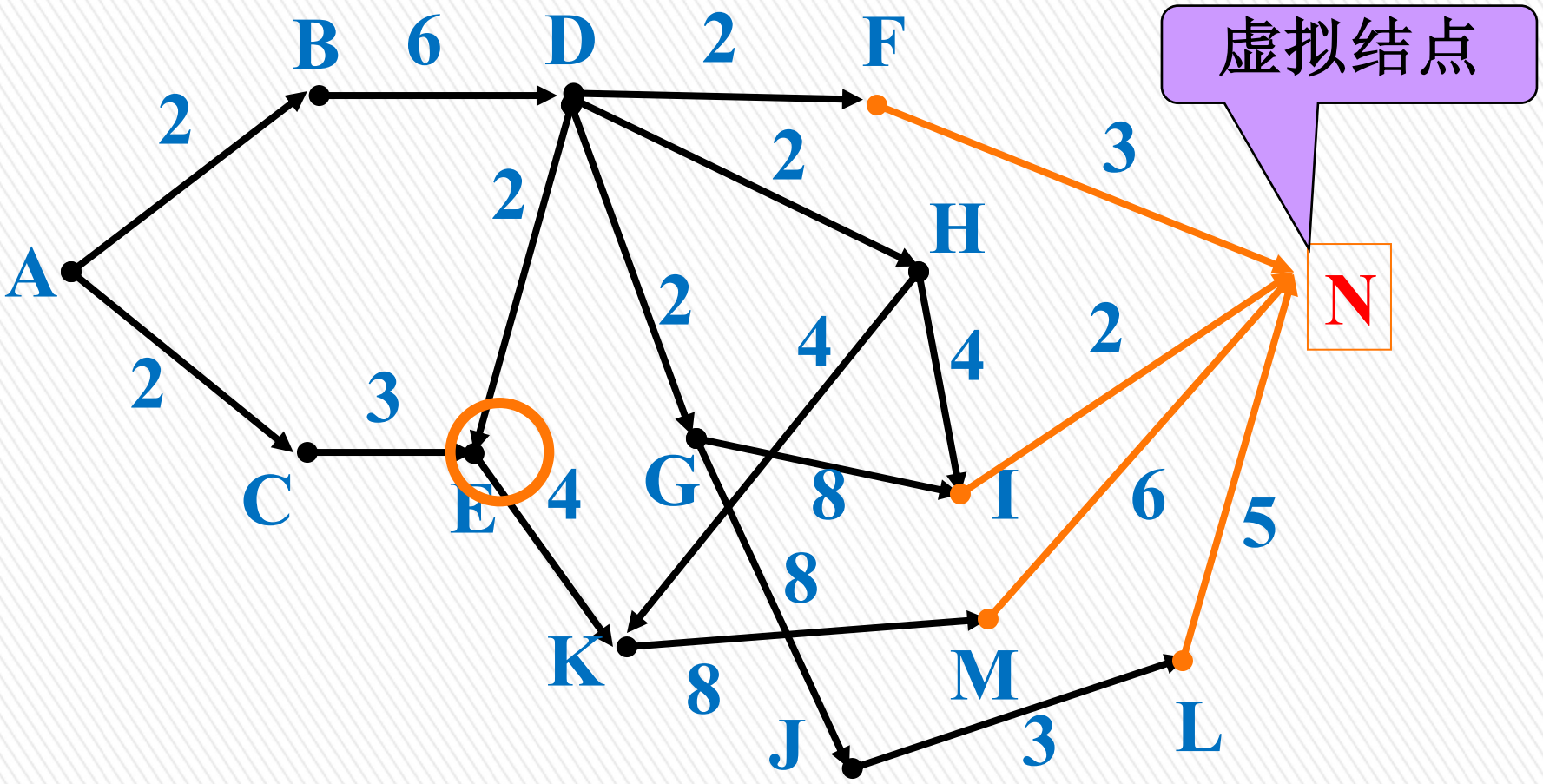
关键路径

例：一项工程由13道工序组成，所需时间(单位：天) 及先行工序如下表所示

工序	A	B	C	D	E	F	G
用时	2	6	3	2	4	3	8
先行工序	-	A	A	B	C,D	D	D
工序	H	I	J	K	L	M	
用时	4	2	3	8	5	6	
先行工序	D	G,H	G	H,E	J	K	

试问这项工程至少需要多少天才能完成？哪些工程不能延误？哪些工程可以延误？最多可延误多少天？

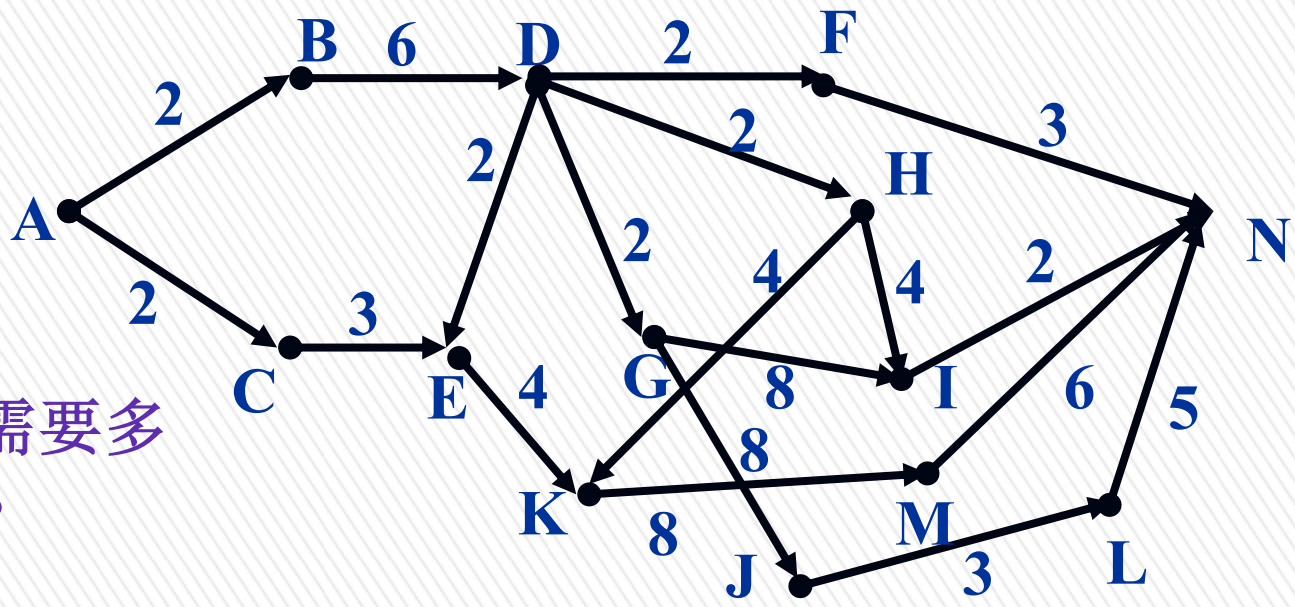
PT图



在PT图中，很容易看出各工序先后完成的顺序及时间。

是否存在多个开始结点？	增设虚拟的超结点来解决

关键路径



这项工程至少需要多少天才能完成?

就是要求A到N的最长路，此路径称为关键路径。

哪些工程不能延误? 哪些工程可以延误? 最多可延误多少天?

关键路径上的那些工程不能延误。

关键路径

引理2.7.1 若有向图G无有向回路，则一定存在入度及出度为0的结点

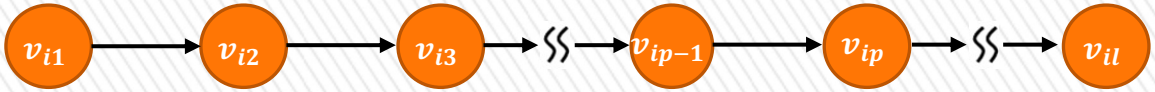
证明：

在G中构造一条极长的有向道路$P(v_{i1}, \dots, v_{il})$，则$d^-(v_{i1}) = 0$，$d^+(v_{il}) = 0$

否则如果$d^-(v_{i1}) \neq 0$，则一定有边$(v_k, v_{i1}) \in E(G)$，此时：

- 若$v_k \in P$，则G存在有向回路，与已知矛盾；
- 若$v_k \notin P$，则P不是极长道路，与假设矛盾。

因此$d^-(v_{i1}) = 0$，同理可证$d^+(v_{il}) = 0$



关键路径

定理2.7.1 若有向图G无有向回路，则可以将G的结点重新编号为$v_1', v_2', \dots, v_n'$，使得对任意的边$v_i'v_j' \in E(G)$，都有$i < j$。

图中所有边都是从编号小的结点指向编号大的结点

证明：

构造法：构造结点序列

由引理2.7.1，G中存在入度为0的点

取一个这样的点编号为v_1'，令$G \leftarrow G - v_1'$，

得到的图仍然没有回路

再取一个入度为0的点编号为v_2'，

重复这个过程，直到所有点都编号为止

这时所有的编号都满足定理的条件

拓扑排序

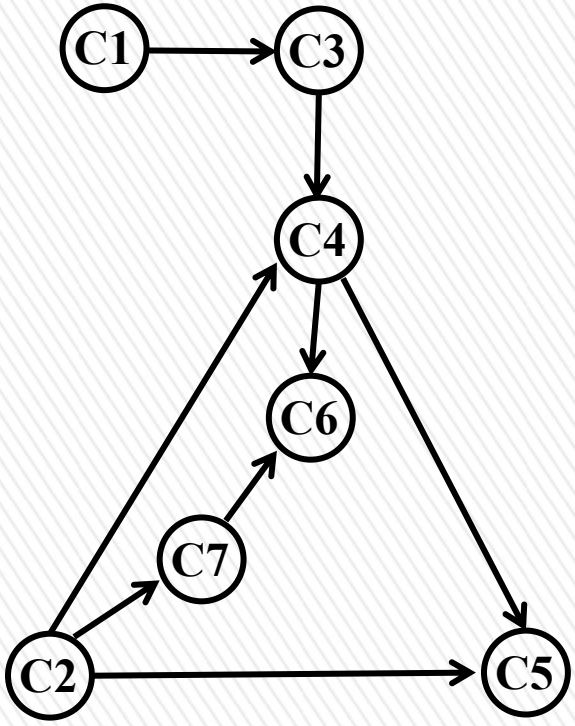
定义：设$G=(V,E)$是一个具有n个顶点的有向图，V中顶点序列$v_1, v_2, \ldots, v_n$称为一个拓扑序列，当且仅当该顶点序列满足下列条件：若$<v_i, v_j>$是图中的边(即从顶点v_i到v_j有一条路径)，则在序列中顶点v_i必须排在顶点v_j之前。

在一个有向图中找一个拓扑序列的过程称为拓扑排序。

拓扑排序

例2.7.1：计算机专业的学生必须完成一系列规定的基础课和专业课才能毕业，假设这些课程的名称与相应代号有如下关系：

代号	课程名称	先修课程
C1	高等数学	无
C2	程序设计	无
C3	离散数学	C1
C4	数据结构	C2, C3
C5	编译原理	C2, C4
C6	操作系统	C4, C7
C7	计算机组成原理	C2



拓扑排序

算法：

(1) 从有向图中选择一个没有前驱(即入度为0)的顶点并且输出它。

(2) 从图中删去该顶点，并且删去从该顶点发出的全部有向边。

(3) 重复上述两步，直到剩余的图中不再存在没有前驱的顶点为止。

如果还有顶点却没有入度为0的顶点，说明有向图有环存在。

拓扑排序

为每个顶点设立一个链表，每个链表有一个表头结点，这些表头结点构成一个数组，表头结点中增加一个存放顶点入度的域***count***：

```
typedef struct                  /* 表头结点类型 */
{       Vertex data;            /* 顶点信息 */
        int count;              /* 存放顶点入度 */
        ArcNode *firstarc;      /* 指向第一条弧 */
} VNode;
```

拓扑排序

```
void  TopSort (VNode adj[], int n)
{    int i, j; int St[MAXV], top = -1;       /*栈St的指针为top*/
     ArcNode *p;
     for (i = 0; i < n; i ++)
          if (adj[i].count == 0)             /*入度为0的顶点入栈*/
          {    top ++; St[top] = i;    }
     while (top > -1)                        /*栈不为空时循环*/
     {    i = St[top]; top --;               /*出栈*/
          printf ("%d ", i); p = adj[i].firstarc;
          while (p != NULL)
          {    j = p->adjvex;
               adj[j].count --;
               if (adj[j].count == 0)
               {    top ++;  St[top] = j;  }
               p = p->nextarc;               /*找下一个相邻顶点*/
          }
     }
}
```

多选题 1分

设置

以下哪些关于拓扑排序的论述是正确的?

A 拓扑排序只能应用于有向无环图

B 对于一个给定的图，拓扑序可能不唯一

C 如果序列X是图G的拓扑序，则“序列X的逆序”是“G的转置图”的拓扑序

提交

关键路径

- 关键路径
  - 当节点重新编号以后，设v_1到各点最长路径长度分别是$0 = \pi(1'), \pi(2'), \ldots, \pi(n')$
  - 则： $\pi(j') = \max_{0 < i' < j'} (\pi(i') + l_{i'j'})$
  - 这就是关键路径算法的思想。
  - 其中$\pi(i')$就是工序i'的最早启动时间

关键路径

- 关键路径算法

① 根据定理2.7.1对结点重新编号为$v_1', v_2', \cdots, v_n'$

② 赋初值$\pi(v_1') = 0$

③ 依次更新$\pi(v_j'), j = 2, 3, \cdots, n$

$$\pi(v_j') = \max_{v_i' \in \Gamma^- v_j'} (\pi(v_i') + w(v_i', v_j'))$$

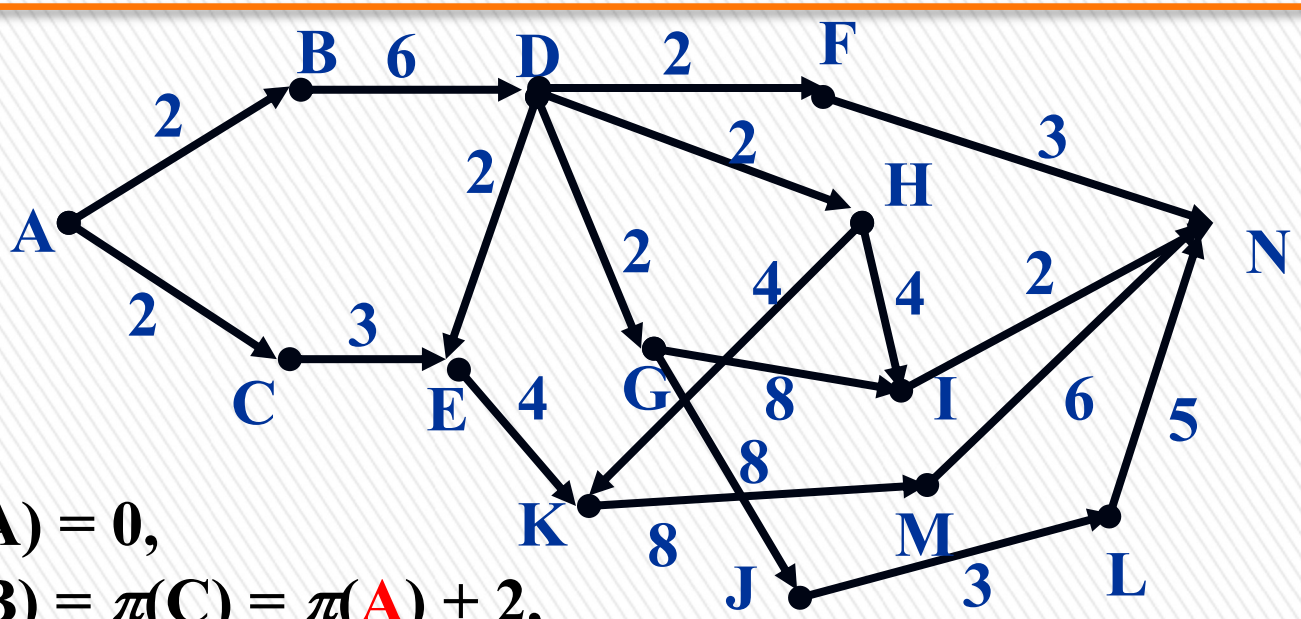
④ 结束

算法复杂性分析：

步骤1：m次减法和判断；

步骤2：m次加法和比较总的计算复杂度：$O(m)$

关键路径实例



$\pi(A) = 0,$

$\pi(B) = \pi(C) = \pi(A) + 2,$

$\pi(D) = \pi(B) + 6,$

$\pi(E) = \max\{\pi(C) + 3, \pi(D) + 2\} = 10,$

$\pi(F) = \pi(G) = \pi(H) = \pi(D) + 2 = 10,$

$\pi(I) = \max\{\pi(G) + 8, \pi(H) + 4\} = 18,$

$\pi(J) = \pi(G) + 8 = 18,$

$\pi(K) = \max\{\pi(E) + 4, \pi(H) + 4\} = 14,$

$\pi(L) = \pi(J) + 3 = 21,$

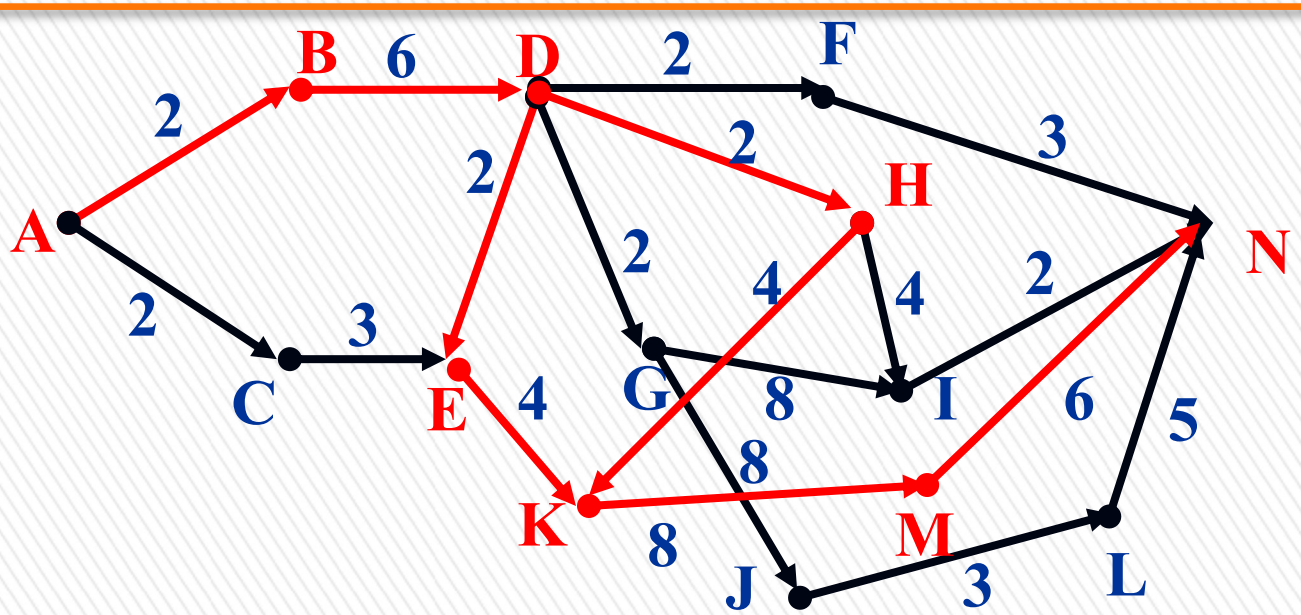
$\pi(M) = \pi(K) + 8 = 22,$

$\pi(N) = \max\{\pi(F) + 3, \pi(I) + 2, \pi(L) + 5, \pi(M) + 6\} = 28.$

拓扑排序.

该例已排好序

关键路径实例



π(N)=28，工程至少需要28天才能完成。

关键路径(最长路径):

A→B→D→E→K→M→N

A→B→D→H→K→M→N

路径上的工序{A, B, D, E, H, K, M}不能延误，否则影响工期

对于不在关键路径上的工序，是否允许延误？如果允许，最多能够延误多长时间呢？

关键路径

- 最大允许延误时间
  - 显然关键路径上的工序是不允许延误的，否则不可能按时完成工程项目
  - 而对非关键工序的允许延误时间将可以给工程规划人员带来工作上的灵活性

各工序允许延误时间$t(v_j')$等于各工序最晚启动时间$\tau(v_j')$减去各工序最早启动时间$\pi(v_j')$。

$$t(v_j') = \tau(v_j') - \pi(v_j')$$

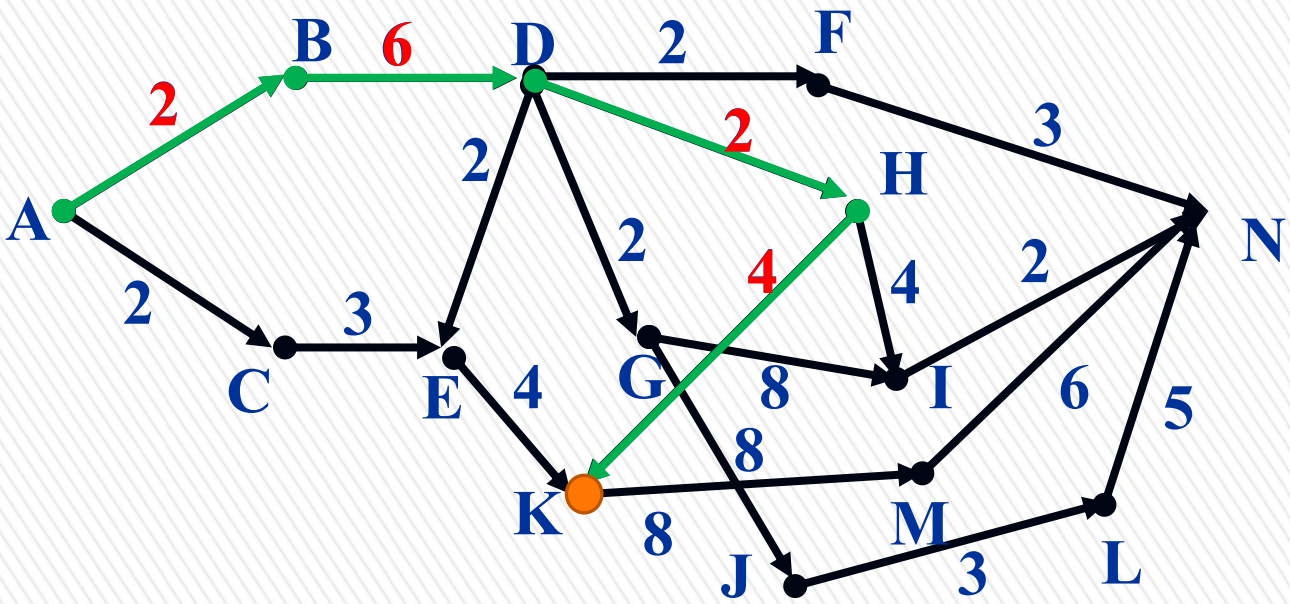
如何确定最晚启动时间？？？

关键路径

- 回顾最早启动时间

对于拓扑序列v_1', v_2', … , v_n',

v_i'的最早启动时间依据的是为v_1'到v_i'的最长路径的长度



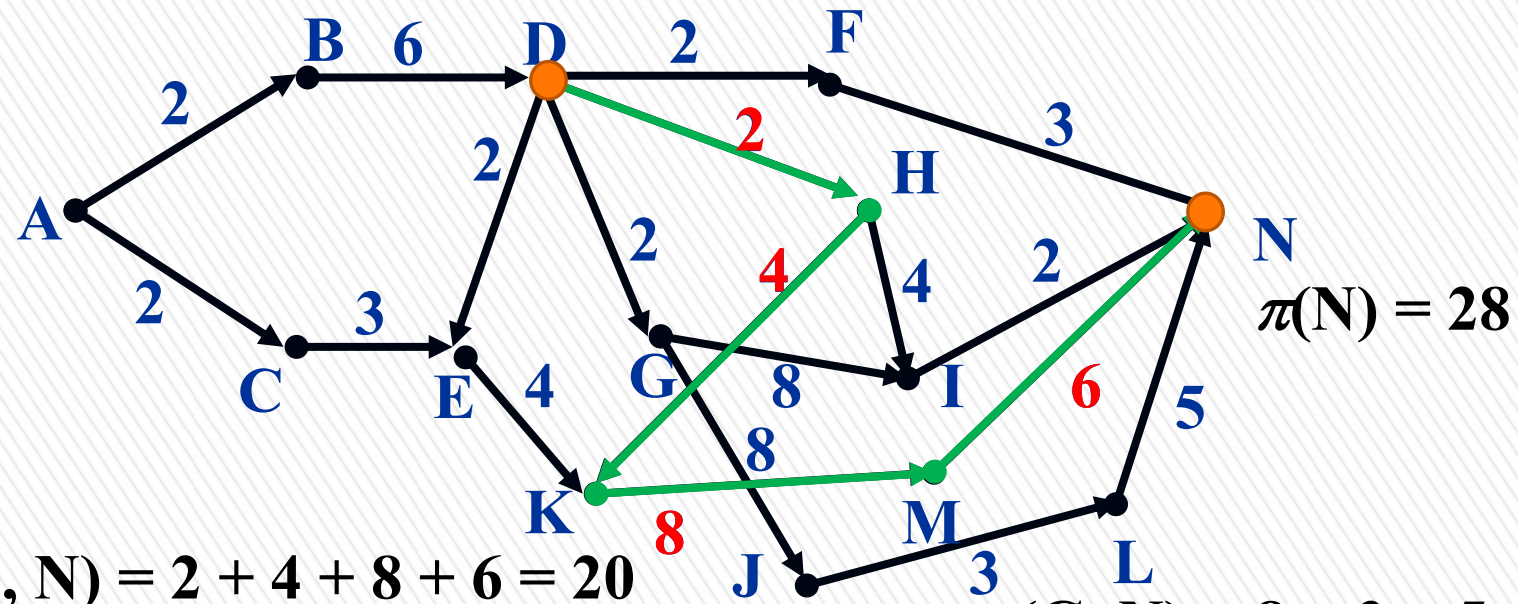
$\pi(K) = 2 + 6 + 2 + 4 = 14$

关键路径

- 相对应，确定最晚启动时间的思路：

对于拓扑序列v_1', v_2', … , v_n',

v_i'的最晚启动时间依据的是为v_i'到v_n'的最长路径的长度



$\pi(N) = 28$

$\pi(D, N) = 2 + 4 + 8 + 6 = 20$

$\tau(D) = \pi(N) - \pi(D, N) = 8$

$t(D) = \tau(D) - \pi(D) = 0$

$\pi(G, N) = 8 + 3 + 5 = 16$

$\tau(G) = 12,\ \pi(G) = 10$

$t(G) = \tau(G) - \pi(G) = 2$

关键路径

- 最大允许延误时间算法

① 根据定理2.7.1对结点重新编号为$v_1', v_2', \dots, v_n'$

② 赋初值 $\tau(v_n') = \pi(v_n')$

③ 依次更新 $\tau(v_j'), j = n-1, \dots, 1$

$$\tau(v_j') = \min_{v_i' \in \Gamma^+(v_j')} (\tau(v_i') - w(v_j', v_i'))$$

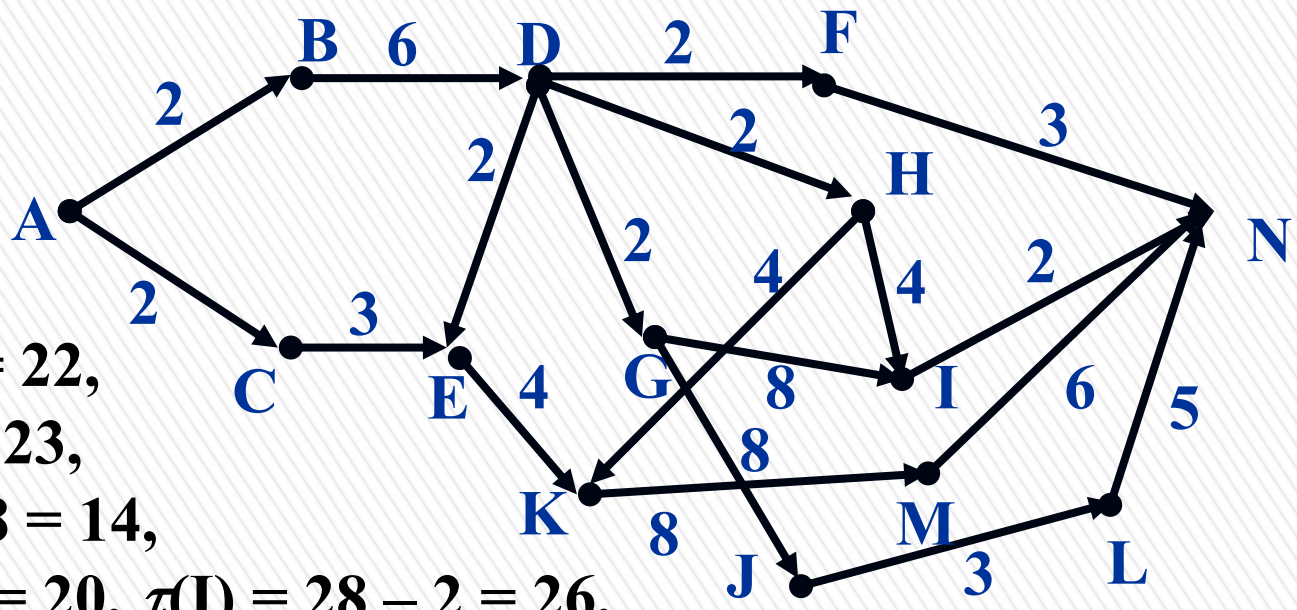
④ 结束

可以看出算法与计算最早启动时间类似

最早启动求“最大值”，最晚启动求“最小值”

这样G中每个结点都具有两个值：最早启动时间和最晚启动时间，两者相减即为该结点对应工序的允许延误时间。

关键路径



$\tau(N) = 28,$

$\tau(M) = 28 - 6 = 22,$

$\tau(L) = 28 - 5 = 23,$

$\tau(K) = \tau(M) - 8 = 14,$

$\tau(J) = \tau(L) - 3 = 20,\ \tau(I) = 28 - 2 = 26,$

$\tau(H) = \min\{\tau(K) - 4,\ \tau(I) - 4\} = 10,$

$\tau(G) = \min\{\tau(J) - 8,\ \tau(I) - 8\} = 12,$

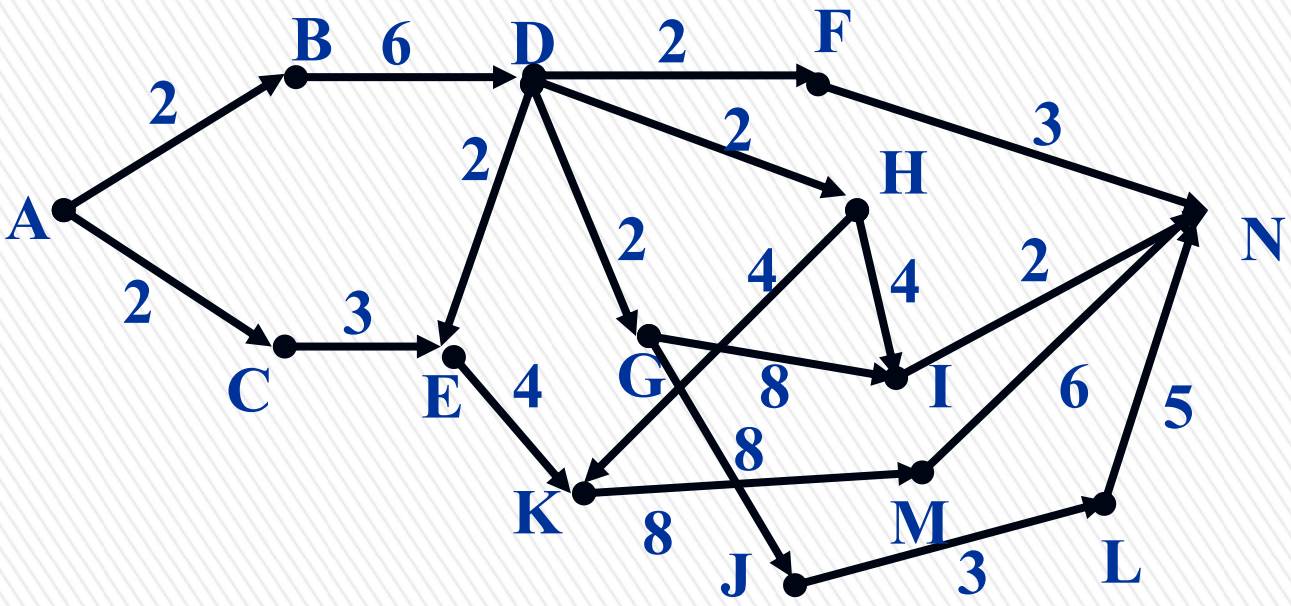
$\tau(F) = 28 - 3 = 25,$

$\tau(E) = \tau(K) - 4 = 10,$

$\tau(D) = \min\{\tau(E) - 2,\ \tau(F) - 2,\ \tau(G) - 2,\ \tau(H) - 2\} = 8,$

$\tau(C) = \tau(E) - 3 = 7,\ \tau(B) = \tau(D) - 6 = 2,\ \tau(A) = 0.$

关键路径



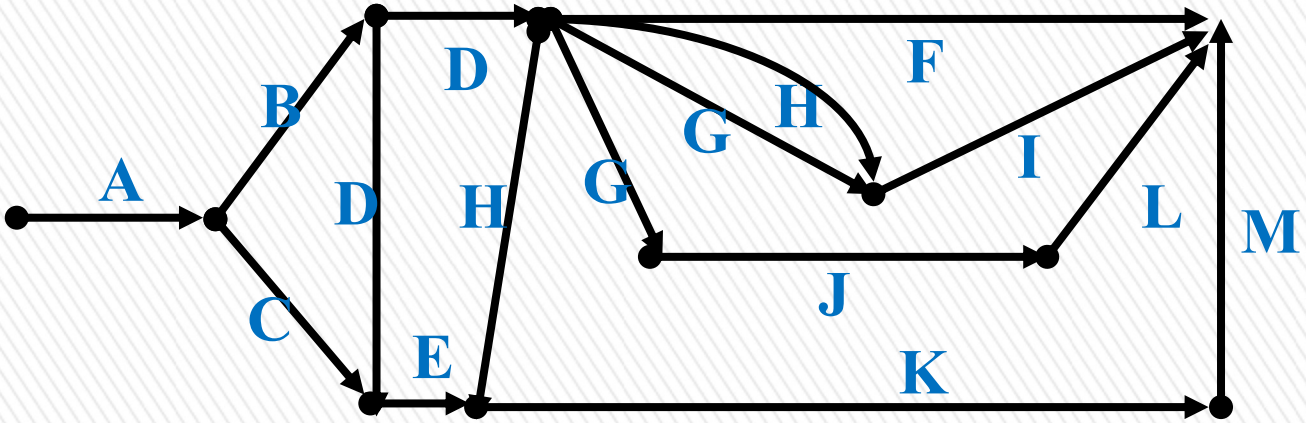
各工序允许延误时间如下：

$t(A) = t(B) = t(D) = t(E) = t(H) = t(K) = t(M) = 0,$

$t(C) = 5, t(F) = 15, t(G) = 2, t(I) = 8, t(J) = 2, t(L) = 2.$

PERT图

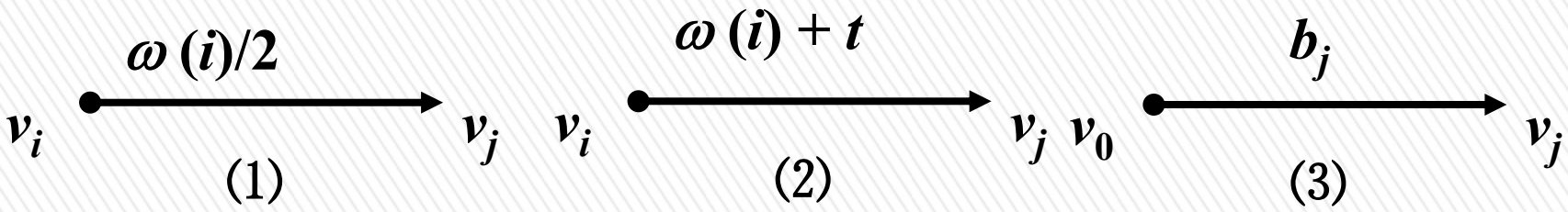
- **PERT图(Programme evaluation and review technique)**
  - 结点为工序之间的关系
  - v_k是e_i的终点、e_j的始点表示工序e_i完成后e_j才能开始
  - 有向边表示工序
  - 边权w_i表示该工序所需时间



对于PERT图，一些算法与PT图类似.

PT图与PERT图的比较

- 各具特色
  - PERT的结点和边数更少些，省内存
  - PT图的结点数等于工序数，更加固定，易于编程
  - PT图更加灵活，能适应一些额外的约束



(1) 表示工序v_i完成一半之后v_j就可以开始
(2) 表示工序v_i完成后经过t时刻v_j才开始
(3) 表示在时间b_j之后工序v_j才能开始，其中v_0表示虚拟结点

中国邮路问题

中国邮路(The Chinese postman problem)

- 问题提出　我国管梅谷教授在1960年首先提出

  邮递员从邮局出发，走遍辖区内所有的街道至少一次(可以重复)，最后回到邮局。要走怎样的路线全程才最短？

- 图论模型

  在一个正权连通图*G*中，从某点出发经过每条边至少一次最后返回出发点的最短回路称为最佳邮路（中国邮路）

  这是一个既与Euler图有关又与最短路有关的问题。

中国邮路

- 非欧拉图

G的任何最佳邮路，必然通过某些边将超过一次。

中国邮递员问题就变化为：给出一个连通图G:

(1) 求$E_1 \subseteq E$满足条件: 在G中重复E_1中每条边, 使得到的图G^*是Euler图(称这样的E_1为可行集)。并且使其边权和尽可能小(称这样的可行集E_1为最优集)。

(2) 求G^*的Euler回路。

中国邮路(The Chinese postman problem)

- 欧拉回路？

  当*G*所有结点度都是偶数时，即*G*有欧拉回路，该回路就是最佳邮路

- 当G只有1个结点的度为奇时？

  不存在

- 当G只有2个结点的度为奇时？

  存在一条欧拉道路，该道路加上从起点到终点的最短路径组成的回路就是最佳邮路

- 当奇数度结点的个数大于2时？

  ？

中国邮路

定理2.8.1 L是无向连通图G最佳邮路的充要条件是:

(1) G的每条边最多重复一次.

(2) 在G的任意一个回路上，重复边的长度之和不超过该回路长度的一半

证明：必要性（1. 最佳邮路G中每条边最多重复一次）

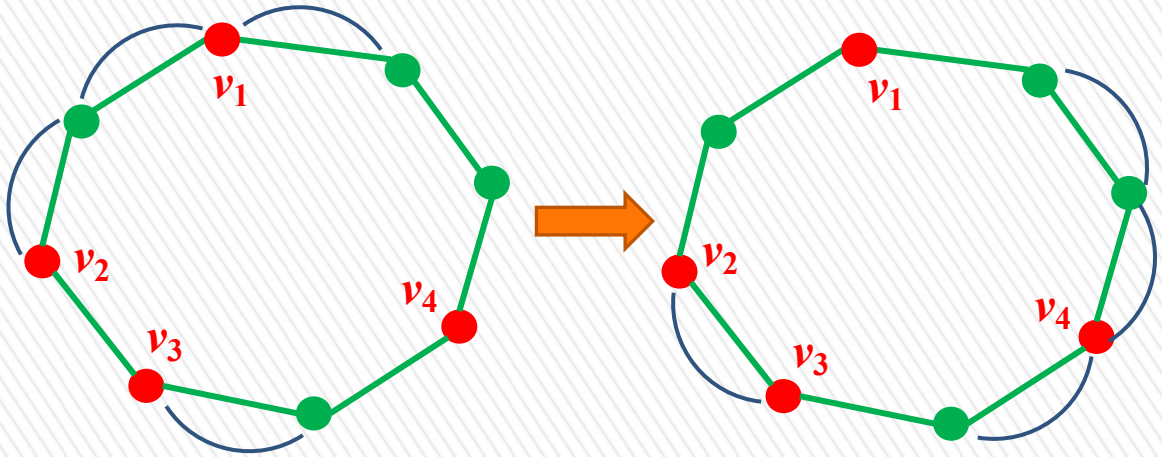
- 若一条最佳邮路重复经过图的某些边，将G中k次重复的边画k次，得到G'。
- 设最佳邮路L'使G中边e_{ij}重复$n(n>1)$次，这时G'中有欧拉回路L'，若使e_{ij}重复$n-2$次，得到G''，G''各点度仍是偶数
- G''的欧拉回路L''也是G的一条中国邮路，且L''长度小于L'，与L'是最佳邮路矛盾。因此边e_{ij}最多重复一次。

中国邮路

证明(续)：

必要性（2. 任一回路重复边长不超过回路长度一半）

- 假设G中某个回路C的重复边的长度大于C总长度的一半
- 令C中重复的边不重复，不重复的边重复，得到G''
- G''仍是欧拉图，因为回路上每个顶点的度数改变0或2，不会改变欧拉图的性质。且L''长度小于L'，与L'是最佳邮路矛盾
- 因此，在任意一个回路上，重复边的长度之和不会超过回路的一半



中国邮路

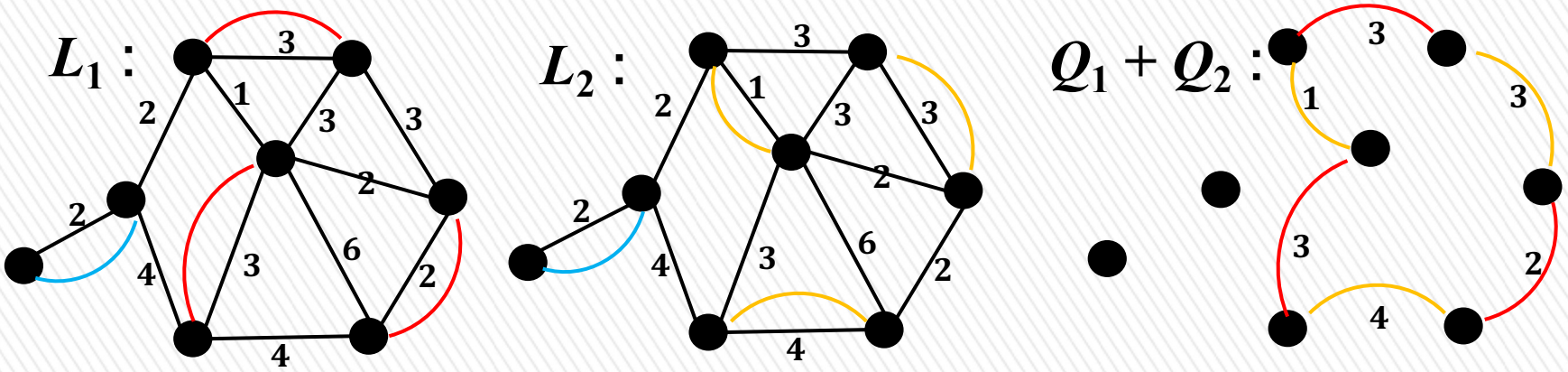
证明(续)：

充分性（无向图满足这两个条件后成为的欧拉图必为最佳邮路）

只需证明凡满足定理中条件(1)和(2)的重复边集，其总长度均相等即可

- 设$L_1 = E(G) + Q + Q_1$，$L_2 = E(G) + Q + Q_2$，其中Q是L_1和L_2共同的重复边集，Q_1和Q_2是分别只属于L_1和L_2的重复边集，L_1和L_2的结点度数均为偶数
- 设对称差$E'(G) = Q_1 + Q_2$， $G' = (V(G), E'(G))$为简单图，可能不连通，但其每个结点的度数均为偶数，（因为$L_1 + L_2$为偶数，减去$(2E(G) + 2Q)$，每个节点度数都减去2的倍数，仍为偶数）

所以图G'可分为若干个回路，所有结点度数均为偶数。



中国邮路

证明(续)：

充分性（无向图满足了这两个条件后必为最佳邮路）

- 故可构造$G' = (V(G), E'(G))$，G'是简单图，各结点度是偶数
- 若$E'(G) = \Phi$，显然$\pi(L_1) = \pi(L_2)$
- 否则（即$E'(G) \neq \Phi$，其中对称差$E'(G) = Q_1 + Q_2$）

  因为G'是简单图，各结点度是偶数，可分为若干个回路

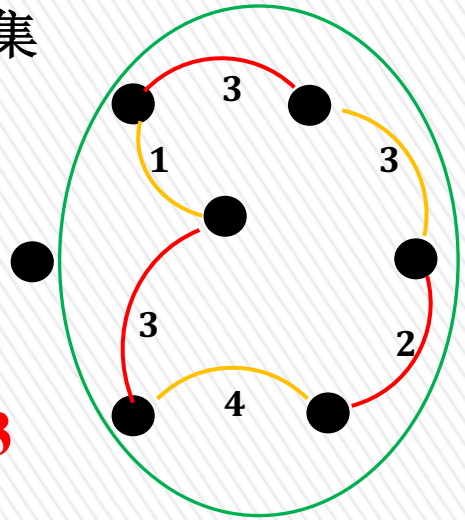
  对任一个回路C，设C_1和C_2分别是L_1和L_2的重复边集

  由已知条件(2)可知，$\pi(C_1) \leq \pi(C_2), \pi(C_2) \leq \pi(C_1)$

  因此$\pi(C_1) = \pi(C_2)$，故$\pi(L_1) = \pi(L_2)$

充分性得证

G'：



$1 + 3 + 4 = 2 + 3 + 3$

定理给出了中国邮路问题的一种算法称为“奇偶点图上作业法”

中国邮路

- 构造中国邮路算法

① 找出度为奇的点

② 依据条件1构造邮路，即G的每条边最多重复一次，并保证计算重复边之后度都是偶数

③ 由条件2对所有回路进行判断，在G的任意一个回路上,如果重复边的长度之和超过该回路长度的一半，则令回路中的重复边不重复，不重复边变为重复

中国邮路

例：求下图的中国邮递员问题的解。

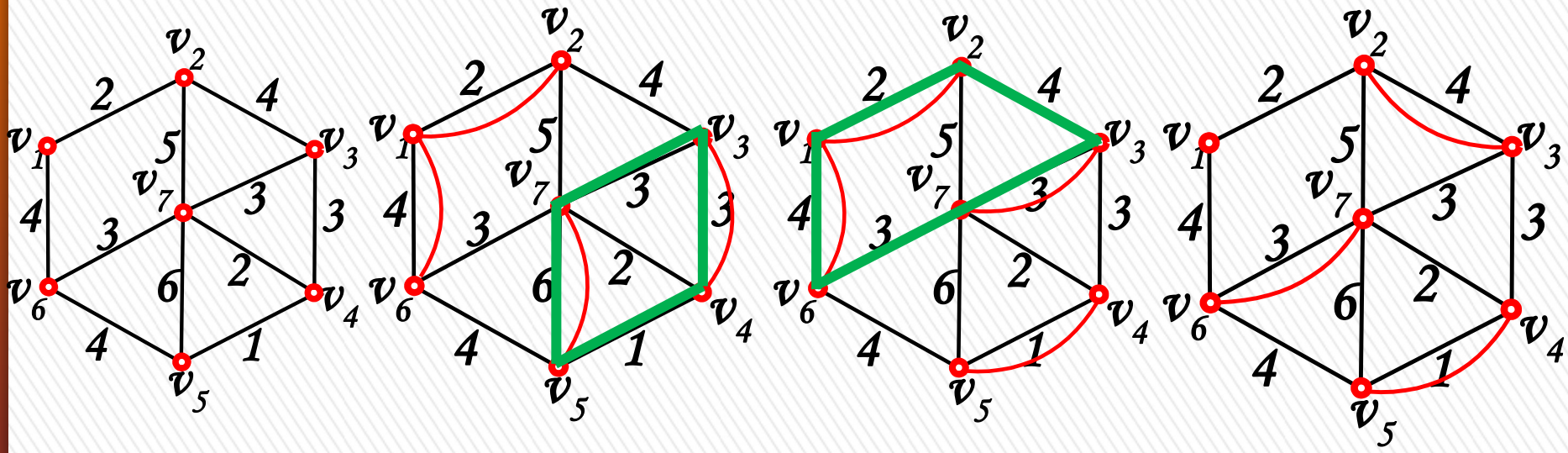
解：(1) 将G变为Euler图

(2) ∵ 在回路$v_3\ v_4\ v_5\ v_7\ v_3$中, 重复边的权>1/2回路的权

∴ 在回路中，重复边不重复，不重复边重复

(3) 在回路$v_1\ v_2\ v_3\ v_7\ v_6\ v_1$中, 重复边的权>1/2回路的权

∴ 同理修改。



中国邮路

- 构造中国邮路算法

  Edmonds最小权匹配算法　　1973年

① 确定G中度为奇的结点，构成$V_0(G)$

② 求$V_0(G)$各结点在G中的最短路径P_{ij}及其长度$\pi(v_i, v_j)$

③ 对$V_0(G)$的结点进行最小权匹配，即选出$|V_0(G)|/2$个$\pi(v_i, v_j)$，保证每个结点在P_{ij}中只出现一次，并且这些$\pi(v_i, v_j)$的总和最小（最佳匹配问题，在后面二分图匹配中会讲如何实现）

④ 在最小权匹配里各$\pi(v_i, v_j)$所对应的路径P_{ij}中的各边在G中重复一次，得到G'

⑤ G'是欧拉图，它的一条欧拉回路即为解

有向图的中国邮路

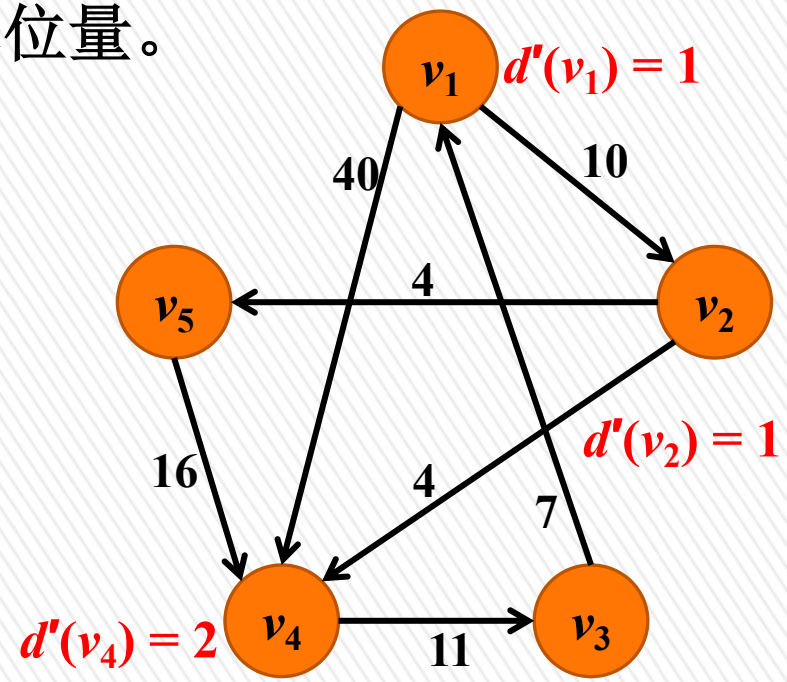
- 有向图可能没有中国邮路
  - 当图中含有正度或负度为0的点时，没有中国邮路
- 若图中各点正负度相同，则存在有向欧拉回路，任一条欧拉回路都是问题的解
- 对于一般的非对称有向图（即存在正负度不相等的结点），如何确定中国邮路？

有向图的中国邮路

- 存在正负度不相等的结点时
  - $d'(v_i) = d^-(v_i) - d^+(v_i)$
  - 若$d'(v_i) > 0$，表示需要$d'(v_i)$次重复经过v_i发出的一些边，或者说v_i可供应$d'(v_i)$个单位量。
  - 若$d'(v_i) < 0$，表示需要$d'(v_i)$次重复经过进入v_i的一些边，或者说v_i可接收$d'(v_i)$个单位量。
  - 若$d'(v_i) = 0$，称v_i是中间结点。

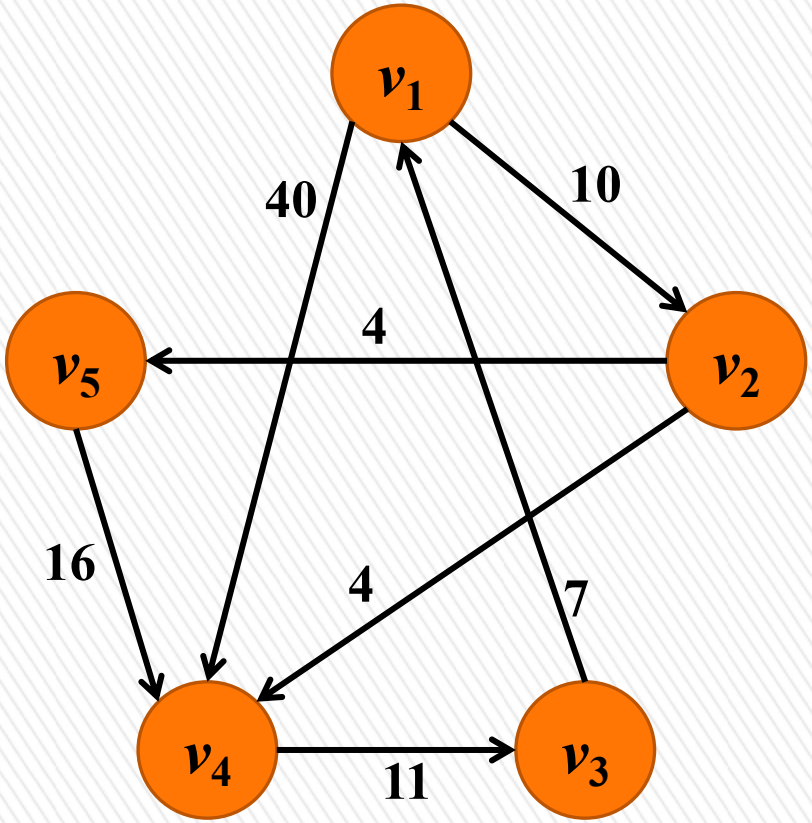
算法设计？

所有结点的d'配对



有向图的中国邮路

例：找出下图中的中国邮路



$d'(v_1) = -1$

$d'(v_2) = -1$

$d'(v_3) = 0$

$d'(v_4) = 2$

$d'(v_5) = 0$

有向图的中国邮路

得到两条总和最小的P_{st}道路

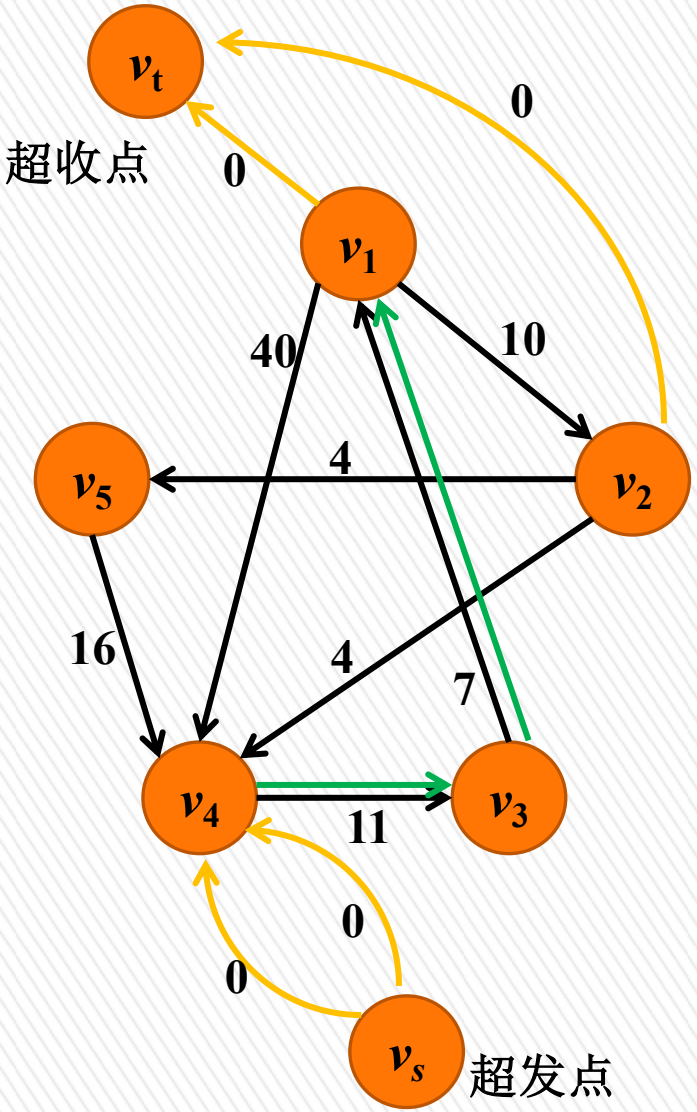
$P_1 = (v_s, v_4, v_3, v_1, v_t)$

$P_2 = (v_s, v_4, v_3, v_1, v_2, v_t)$

边(v_4, v_3)重复2次

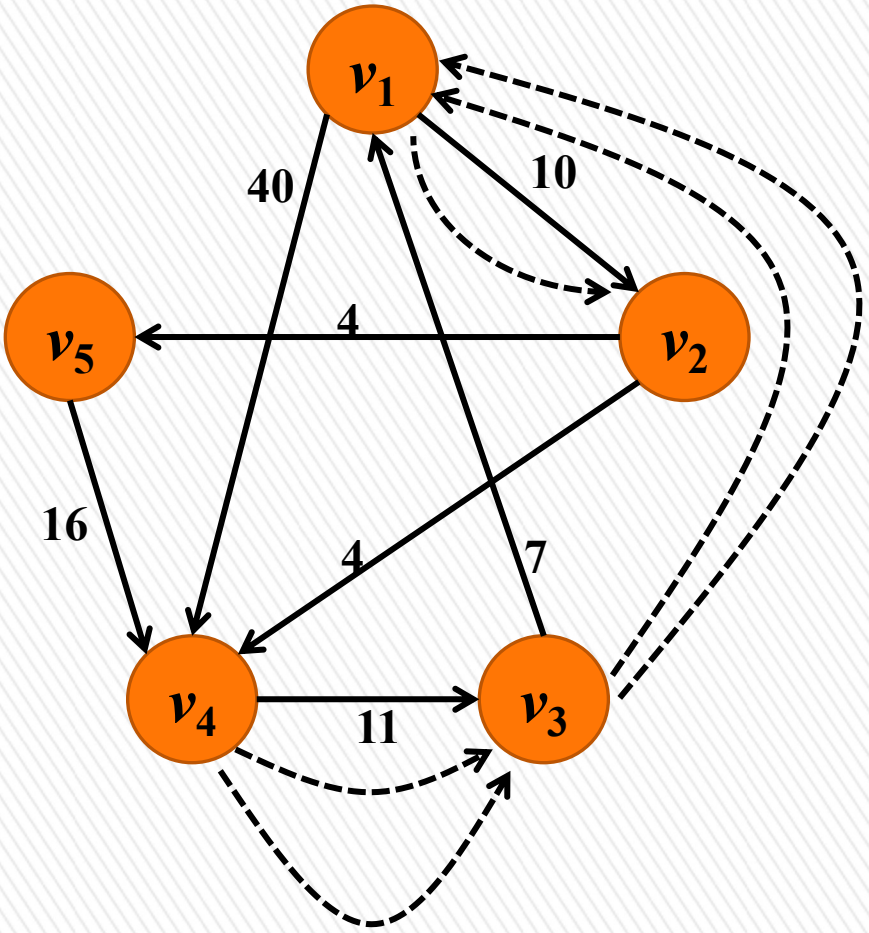
边(v_3, v_1)重复2次

边(v_1, v_2)重复1次



有向图的中国邮路

例：



添加重复边，所构成的图即为欧拉图，其中的一条欧拉回路就是问题的解。

有向图的中国邮路

- 构造有向图的中国邮路算法

① 计算各点的正负度，求出$d'(v_i) = d^-(v_i) - d^+(v_i)$

② 添加一个超发点v_s，对满足$d'(v_i)>0$的结点，加入$d'(v_i)$条有向边(v_s, v_i)，权为0

③ 添加一个超收点v_t，对满足$d'(v_j)<0$的结点，加入$d'(v_j)$条有向边(v_j, v_t)，权为0。得到图G'。

④ 在G'中求$d(v_s)$条过以v_s, v_t为两端点的形如(v_s, v_i)，(v_j, v_t)，每边一次且仅一次的总和最小的P_{st}道路。记下G中各边在这些道路里的重复次数。

⑤ 计入各边的重复次数，G中存在有向欧拉回路，其中一条即为解。

本章总结

- 道路与回路的定义和相关概念
- 道路与回路的判定方法
- 欧拉道路与回路
- 哈密顿道路与回路
- 旅行商问题与分支定界法
- 最短路径
- 关键路径
- 中国邮路

本章总结

- 道路与回路的定义和判定方法
  - 邻接矩阵判定法，Warshall算法、深度、广度优先搜索
- 欧拉、哈密顿道路与回路
  - 判定、构造方法
- 旅行商问题
  - 分支定界法、便宜算法
- 最短路径
  - Dijkstra（正权）、BFS（权为1）、Ford（无负回路）、Floyd算法（任意两点）
- 关键路径
  - PT、PERT图，最早、最晚启动时间和最大允许延误时间
- 中国邮路
  - 无向图、有向图的中国邮路构造方法