图论与代数结构

道路与回路进阶

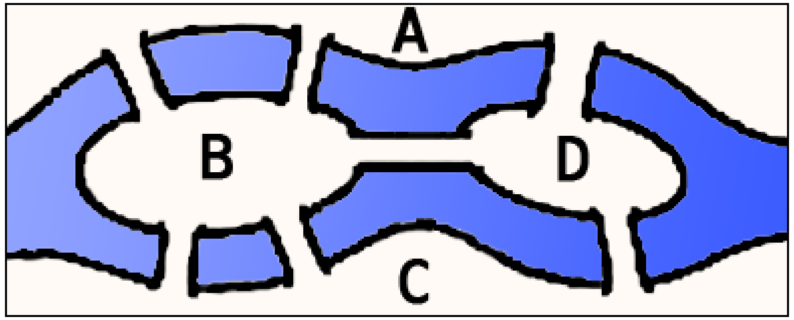
崔 勇
清华大学计算机系
网络技术研究所

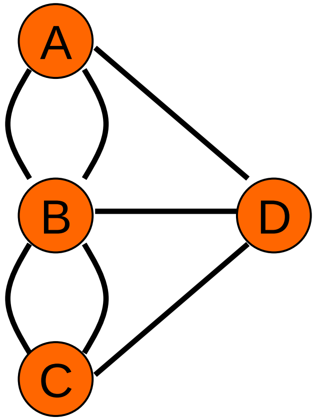
主要内容 道路与回路进阶

- 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

欧拉道路与回路

7桥问题（一笔画问题）：能否从某处出发，经过各桥一次且仅一次，最后返回原处?



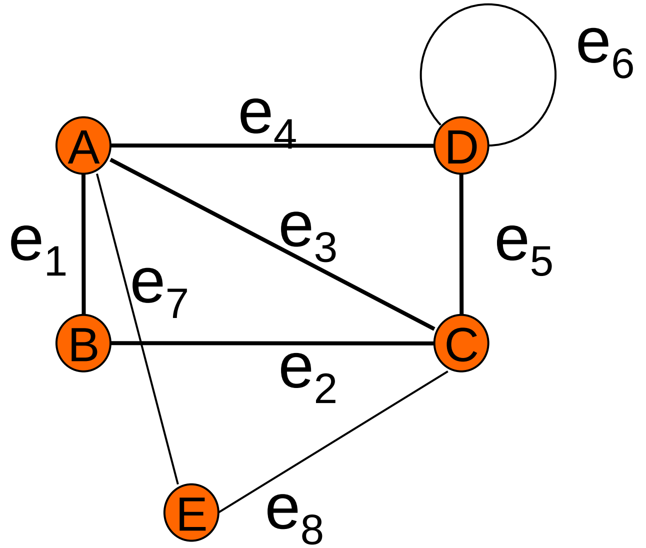


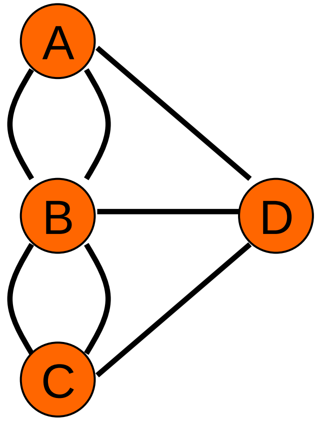
- 欧拉道路（回路）
  - 无向连通图G=(V,E)中的一条经过所有边的简单道路（回路），称为G的欧拉道路（回路）

即问在上图中是否存在欧拉回路?　　　呼唤存在性定理?

欧拉道路与回路(2)

- 定理
  - 无向连通图G有欧拉回路的充要条件是各顶点的度都是偶数。
  - 证明（充要条件？）
    - 必要性
      - 已知存在欧拉回路，要证明度都是偶数
      - 欧拉回路经过每边一次且仅一次
      - 沿该回路进入某点后，必定经由另一条边出去
      - 对每一点的进出次数相同
      - 因此，各点的度都是偶数

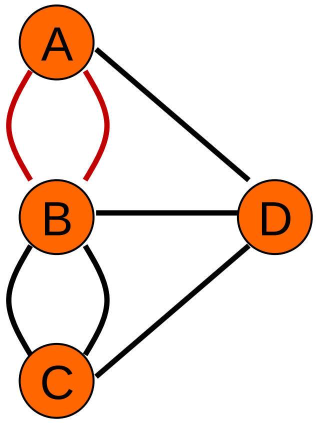




欧拉道路与回路(3)

- 证（续）：
  - 充分性
    - “无向连通图G有欧拉回路的充要条件是各顶点的度都是偶数”（证明思路？）
    - 采用构造法证明

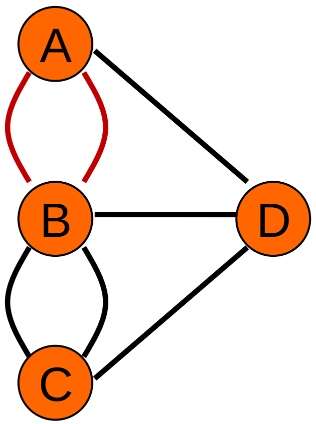
欧拉回路的特点



  - 从任意点v_0出发，构造G的一条简单回路C
    - 由于v_i的度为偶，所以不可能停留在某点$v_i \in V - v_0$上，而不能继续向前构造
    - 由于G是有穷图，因此最终一定能够回到v_0，构成简单回路C
  - 若C包含了G中的所有边，它即是G的欧拉回路

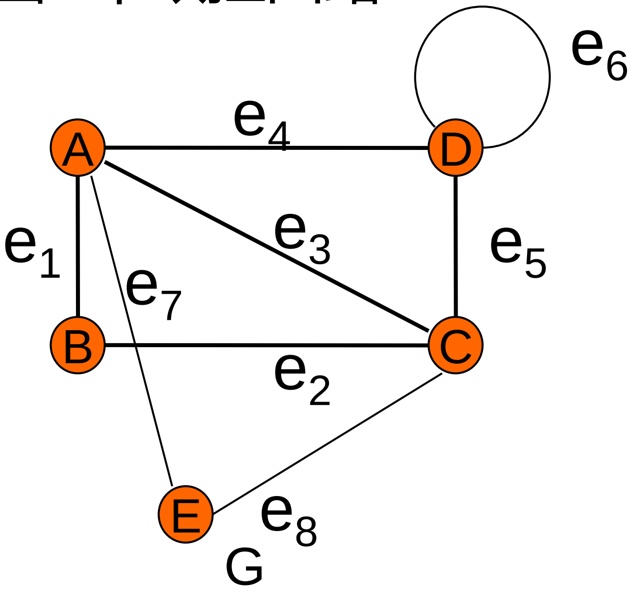
欧拉道路与回路(4)

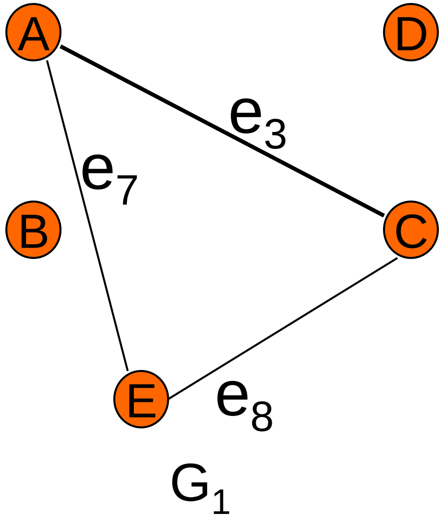
- 证（续）：
  - 否则，从G中删去C的各边，得到G_1 = G-C
  - 显然G_1中每点的度仍然是偶数
  - 此时，G_1中一定存在度非0的顶点v_i，它同时还是回路C经过的顶点（否则G是非连通图）
  - 这时，在v_i所在的G_1的连通支中，同理可构造简单回路C'，令C=C∪C'，得到包含边数比原来更多的简单回路
  - 继续上述构造过程，最终该简单回路必包含了所有边，即构造出了的一条欧拉回路
  - 充分性证毕“无向连通图G有欧拉回路的充要条件是各顶点的度都是偶数”



欧拉道路与回路(5)

- 例
  - 找出下图G中欧拉回路





从任意一点，如A开始，构造简单回路$C=(e_1,e_2,e_5,e_6,e_4)$

$G_1=G-C$中，A，C度非零，且为G中结点

从A开始构造简单回路$C_1=(e_3,e_8,e_7)$

则$C \cup C_1= (e_1,e_2,e_5,e_6,e_4 ,e_3,e_8,e_7)$是G的一条欧拉回路。

欧拉道路与回路(6)

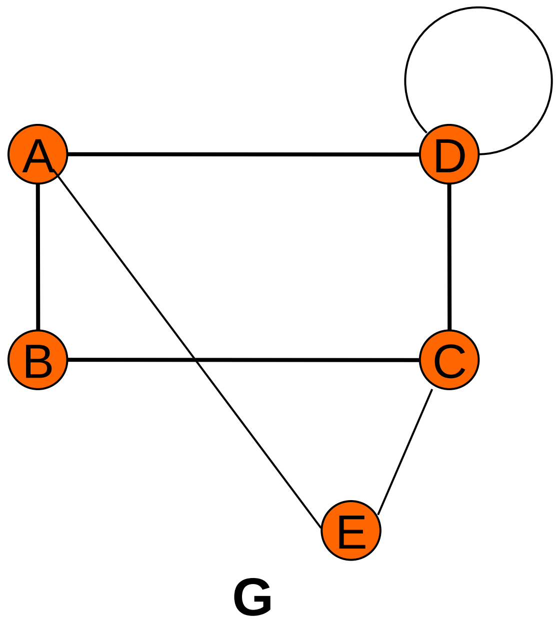
- 全是偶度则欧拉回路，那存在奇度呢？
- 推论（欧拉道路存在性）
  - 若无向连通图G中只有两个奇顶点，则G存在欧拉道路。
  - 证明（思路？）
    - 构造法
    - 设这两个奇顶点是v_i，v_j，
    - 在图G中加入一条边(v_i, v_j)，则所有的顶点的度都为偶，此时其中必然存在一条欧拉回路。
    - 然后将边(v_i, v_j)去掉，可得从v_i到v_j的欧拉道路。

欧拉回路存在性，
然后呢？
什么情况不存在？

单选题 1分

设置

无向连通图G如下所示，则图G：



A 没有欧拉道路

B 有欧拉道路但没有欧拉回路

C 有欧拉回路

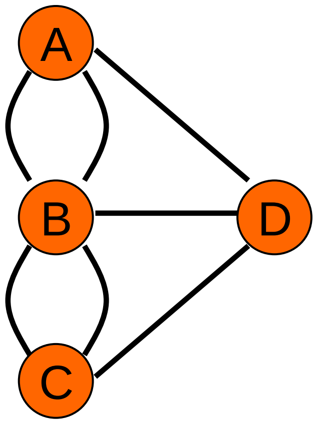
提交

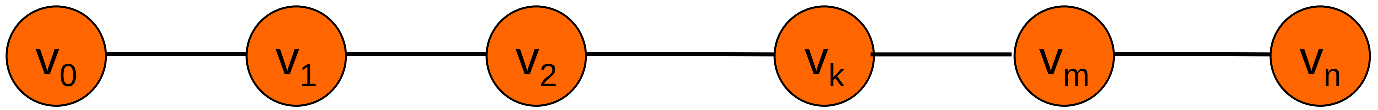
欧拉道路与回路(7)

- 例
  - 设连通图中有K个度为奇数的顶点。证明E(G)可以划分成K/2条简单道路
  - 证明（基本思路？）
    - 构造法
    - 由图的性质可得，K是偶数
    - K个顶点两两配对，增添K/2条边，得到G′
    - G′中每点的度都是偶数，由定理，G′中有欧拉回路C
    - 在C中删去这K/2条边,便得到了K/2条简单道路，它们包含了原图G中的所有边

互不相邻

是K/2条吗?





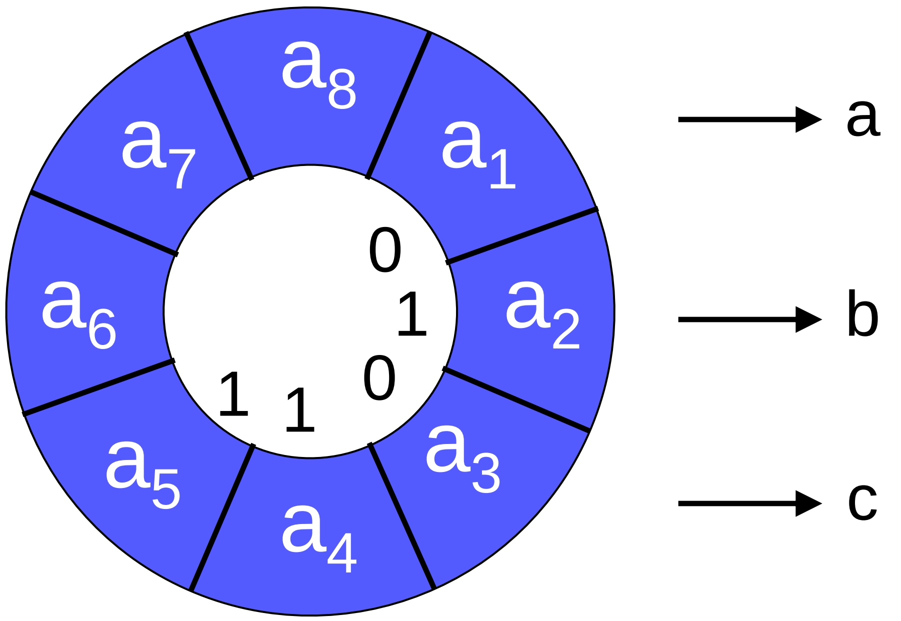
- 即这K/2条简单道路就是E(G)的一个划分

欧拉道路与回路(8)

- 研究了无向图欧拉回路/道路的存在性，那有向图呢?
- 推论
  - 若有向连通图G中各个结点的正度与负度相等，则G中存在有向欧拉回路。
  - 证明
    - 略

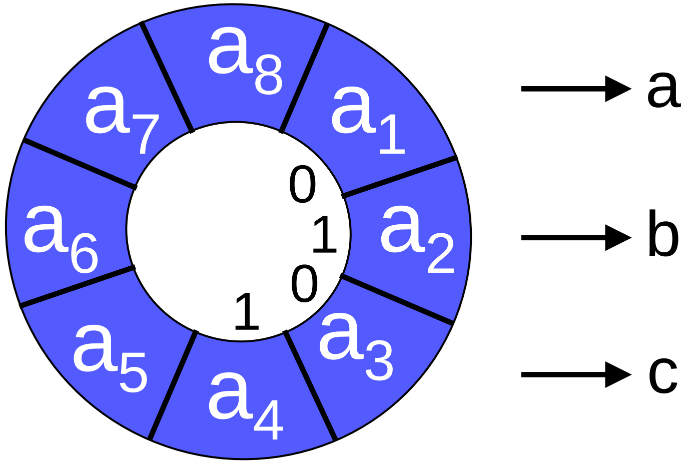
欧拉道路与回路(9)

- 例
  - 如右图，一个编码盘分成8个扇面，每个表示1或者0，其中a，b，c三个位置组成一组输出
  - 当圆盘按照逆时针旋转一格的时候，就会产生一组输出
  - 试问，圆盘上的数怎么排列，可以使圆盘旋转一周能不重复的输出000 ~ 111这8个二进制数(不需要按顺序)?



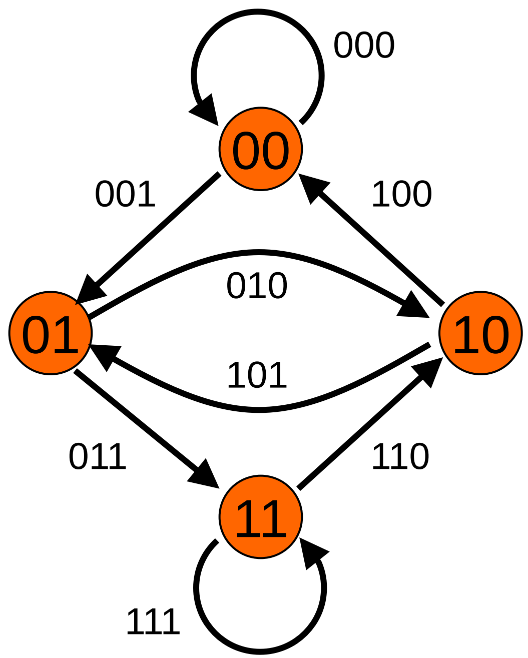
欧拉道路与回路(10)

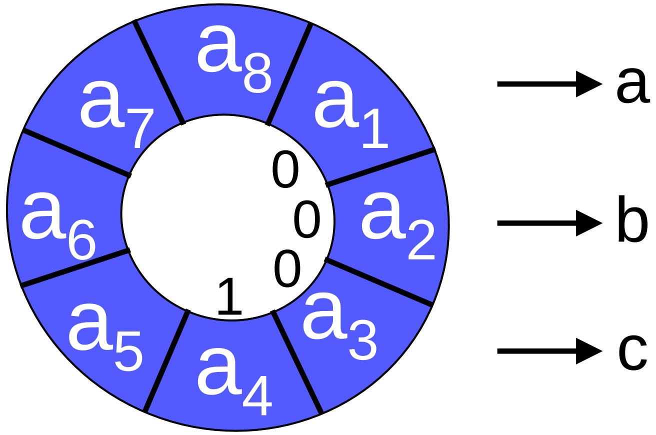
- 例（续）
  - 如何进行建模? 动态!
  - 每次旋转时，输出中有两位不变，如abc变成bcd
  - 结点: 三位数字的前两位
    - 如这里的abc中的ab， bcd中的bc
    - 用0/1组成前两位ab，四种组合情况作为四个结点
  - 边: 结点数字之间的变化关系
    - 每次旋转可以从一个结点ab到另一个结点bc
  - 有两种可能的旋转变化：c = 0 / 1
    - ab输出为ab1或者ab0，即下一状态为b0或b1



欧拉道路与回路(11)

- 例（续）
  - 画出这种转换关系图
    (结点为a_1/a_2位置的数)





1. 八条边表示八个输出值
2. 每个结点的度都是偶数，因此存在欧拉回路
3. 任何一条欧拉回路都是一种可行方案
4. 例如：(上)上左右左 下下右上
5. 所有的a3构成序列"01011100"

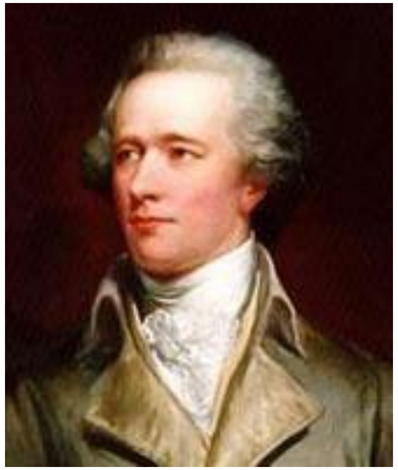
哈密顿道路与回路(1)

- 欧拉道路（回路）
  - 无向连通图中的一条经过所有边的简单道路（回路）称为欧拉道路（回路）
- “清华道路（回路）？”：过所有边/点的简单/初级道路?
- 哈密顿道路（回路）
  - 哈密尔顿的周游世界问题：1857年爱尔兰数学家哈密尔顿发明了“周游世界”玩具，用一个正十二面体的20个顶点表示世界上20个大城市，30条棱代表这些城市之间的道路。要求游戏者从任意一个城市(即顶点)出发，延棱行走经过每个城市一次且只经过一次，最终返回出发地。
  - 无向图连通图的一条过全部结点的初级道路（回路）称为哈密顿(Hamilton)道路（回路）（H-道路，H-回路）

- 哈密顿图：含有 H-回路的图

哈密顿

- 1805年生于爱尔兰都柏林
- 1823-24年间完成多篇几何学和光学的论文
- 年仅22岁的哈密顿被任命为敦辛克天文台的皇家天文研究员和三一学院的天文学教授
- 1834年，哈密顿发表了历史性论文“一种动力学的普遍方法”，成为动力学发展过程中的新里程碑


威廉·罗恩·哈密顿

- 在1843年正式提出了四元数(quaternion)，这是代数学中一项重要成果
- 1836年，皇家学会因他在光学上的成就而授予皇家奖章
- 哈密顿家庭负担很重，为减轻父亲经济压力……
- 发表的论文一般都很简洁，别人不易读懂，但手稿却很详细，因而很多成果都由后人整理而得

哈密顿道路与回路(2)

- 哈密顿回路的研究范畴
  - H回路是初级回路
  - 将任一图中的重边与自环去掉，得到的简单图的H回路的存在性与原图等价
  - 因此，一般考虑简单图…

如何进一步研究?

H道路存在性?

哈密顿道路与回路(3)

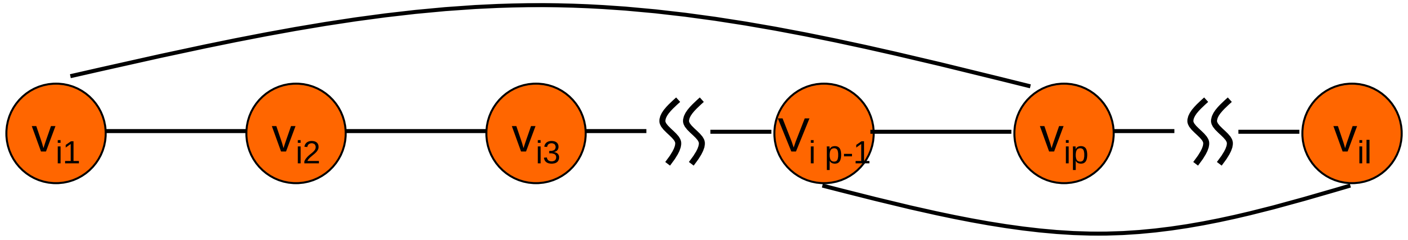
- 定理(什么情况下更可能存在H道路?)
  - 若简单图G中任两点u,v, 恒有 $d(v)+d(u)\geq n-1$, 则G中存在Hamilton道路
  - 证明（基本思路？）：
  - 基本思路：构造法

道路不断加点?

  - (1)证G连通（思考） (2)构造G中的H道路
  - 设P为G的长为l的极长初级道路, P=(v_{i1},v_{i2},…v_{il})，则与v_{i1}和v_{il}相邻的点都在P上
  - 若l=n,则P为H道路

哈密顿道路与回路(4)

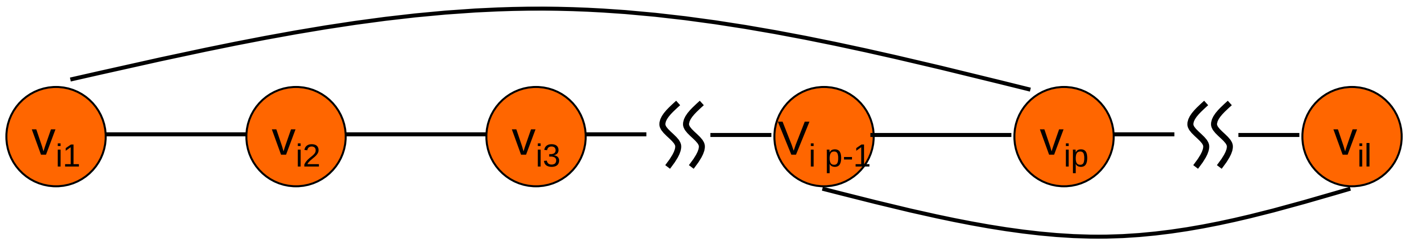
- 证（续）
  - 若$l<n$, 可证明G中一定存在经过结点$v_{i1},v_{i2},\ldots v_{il}$的初级回路C



  - 假设不存在初级回路
  - 设$(v_{i1},v_{ip}) \in E(G)$，则不能有$(v_{i,p-1},v_{il}) \in E(G)$，否则删除$(v_{i,p-1},v_{ip})$，上图形成一个回路
  - 设$d(v_{i1})=k$, 则v_{il}至少与这k个点的左邻居不能相邻，即$d(v_{il}) \le l-k$，考虑v_{il}本身无自环，则$d(v_{il}) \le l-k-1$
  - 则$d(v_{i1})+d(v_{il}) \le l-1 < n-1$，与已知$d(v)+d(u) \ge n-1$ 矛盾
  - 因此$l<n$, 则存在回路C

哈密顿道路与回路(5)

- 证（续）
  - 若$l<n$, 已证明G中一定存在经过结点$v_{i1},v_{i2},\ldots v_{il}$的初级回路C。

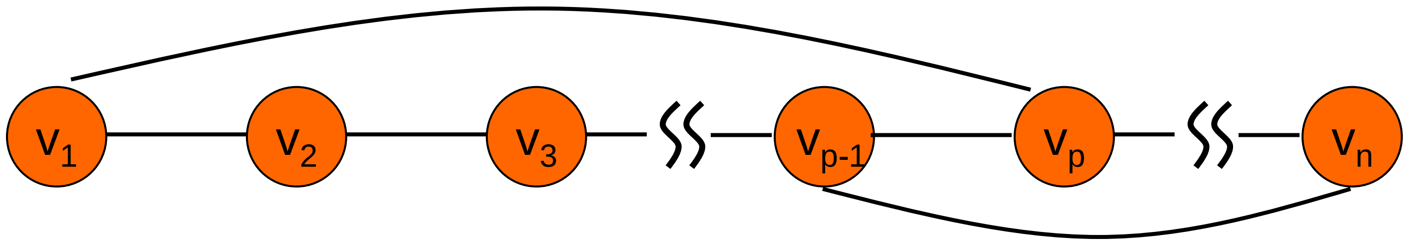


  - 设$C=(v_{i1},v_{i2},\ldots v_{il},v_{i1})$, 由于G连通，故存在C之外的结点$v_t$，必然与C中的某点$v_{iq}$相邻
  - 可构造长为$l+1$的初级道路 $P=(v_t,v_{iq},v_{iq+1},\ldots v_{il},v_{i1}\ldots v_{iq-1})$,
  - 如此构造直到$l=n$,从而P为H道路

H道路存在性充分条件：$\boxed{d(v)+d(u)\geq n-1}$

哈密顿道路与回路(6)

- 推论
  - 如简单图G中任两点u,v, 恒有 $d(v)+d(u)\ge n$， 则G中存在Hamilton回路。
    - 由定理可知，G中存在哈密顿道路，设H为$v_1,v_2,\ldots,v_n$



    - 假设H不是回路
    - 设$d(v_1)=k$, 则$d(v_n)\le n-k-1$（v_n无自环）
    - 则$d(v_n)+d(v_1)\le n-1<n$，与已知矛盾
    - 因此存在初级回路C，即H回路
  - 如简单图G中任意点v, 恒有 $d(v)\ge \frac{n}{2}$， 则G中存在Hamilton回路。

自己能发明出来吗?

哈密顿道路与回路(7)

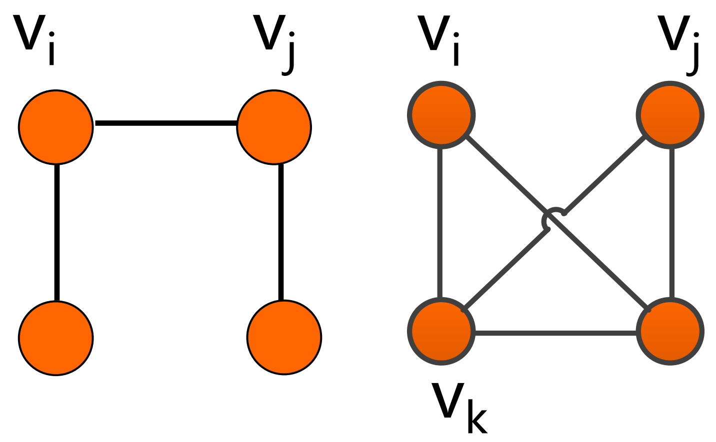
- 例
  - n（n>2）个人中，设任意两人合在一起能认识其余n-2个人，则他们可以站成一排，使相邻者相识
  - 证明
    - 图论建模
    - 每个人看成一个结点
    - 人之间的相识关系作为边
    - 那么以上问题就是这样构造出来的图中存在一条哈密顿道路
    - “任意两人合在一起能认识其余n-2个人”？

哈密顿道路与回路(8)

- 证（续）：
  - 根据题意，任意两个结点v_i, v_j,合在一起能认识其余n-2个人，即有 $d(v_i)+d(v_j)\ge n-2$
  - 若v_i, v_j认识，则有 $d(v_i)+d(v_j)\ge n$;
  - 若v_i, v_j不认识，则任意v_k必同时认识v_i和v_j
    - 否则v_i, v_k合起来不认识v_j，与已知矛盾
    - 因此由n>2，除v_i, v_j外至少还有1人v_k，即存在1人v_k 同时认识v_i, v_j

$$d(v_i)+d(v_j)\ge n-1$$

  - 根据定理，存在哈密顿道路



哈密顿道路与回路(9)

- 引理2.4.1

呼唤：H回路存在性的充要条件

$$d(v)+d(u)\geq n$$ 任意v?

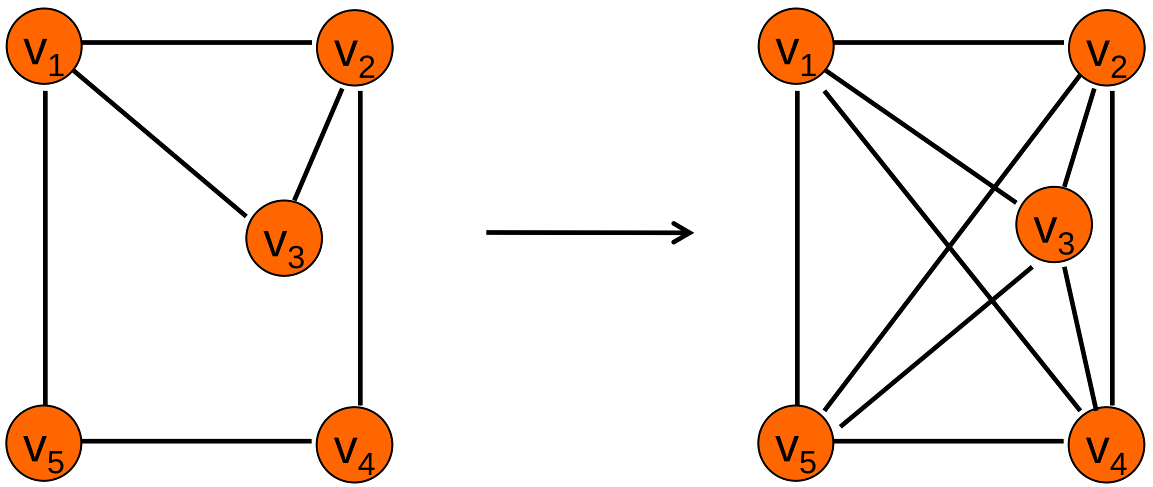
  - 简单图G中v_i和v_j不相邻，且$d(v_i)+d(v_j)\geq n$，则G存在H回路的充要条件是$G+(v_i,v_j)$有H回路
  - 证：（必要性显然，下面证明充分性）
    - 由$G+(v_i,v_j)$存在H回路，考虑H回路是否包含边(v_i,v_j)，若不包含则已得证
    - 若包含(v_i,v_j)，删去此边(v_i,v_j)，G中存在以v_i,v_j为端点的H道路
    - 根据条件$d(v_i)+d(v_j)\geq n$，则H道路可以扩展为H回路，G存在H回路
  - 充分性得证。

能否不断增加这样的边?

给出定义！

哈密顿道路与回路(10)

- 闭合图
  - 若v_i和v_j是简单图G的不相邻结点，满足$d(v_i)+d(v_j) \geq n$，则令$G' = G+(v_i,v_j)$，对G'重复上述过程，直到不再有这样的结点对。最终得到的图称为G的闭合图，记为C(G)



存在性，唯一性?

哈密顿道路与回路(11)

- 引理2.4.2
  - 简单图G的闭合图C(G)是唯一的
  - 证明
    - 设C_1(G)和C_2(G)是G的两个闭合图
    - $L_1=\{e_1,e_2,\ldots,e_r\}$, $L_2=\{a_1,a_2,\ldots,a_s\}$是$C_1$(G)和$C_2$(G)中新加入的边集合
    - 需证$L_1=L_2$
    - 假设$L_1\neq L_2$，为不失一般性，假设$e_{i+1}=(u,v)\in L_1$是构造时第一条不属于L_2的边，令$H=G\cup\{e_1,e_2,\ldots,e_i\}$，则H是$C_2$(G)的子图
    - 由于构造L_1时加入了e_{i+1}，则H有$d(u)+d(v)\geq n$，但是 $(u,v)\notin C_2(G)$ ，与C_2(G)是G的闭合图矛盾

哈密顿道路与回路(12)

- 定理
  - 简单图G存在哈密顿回路的充要条件是其闭合图存在哈密顿回路
  - 证明：
    - 设$C(G)=G\cup L$，$L=\{e_1,e_2,\ldots,e_t\}$
    - 由引理2.4.1和2.4.2
    - G有H回路 $\Leftrightarrow G+e_1$有H回路$\Leftrightarrow\ldots\Leftrightarrow G\cup L$有H回路
    - 由于C(G)唯一，定理得证。

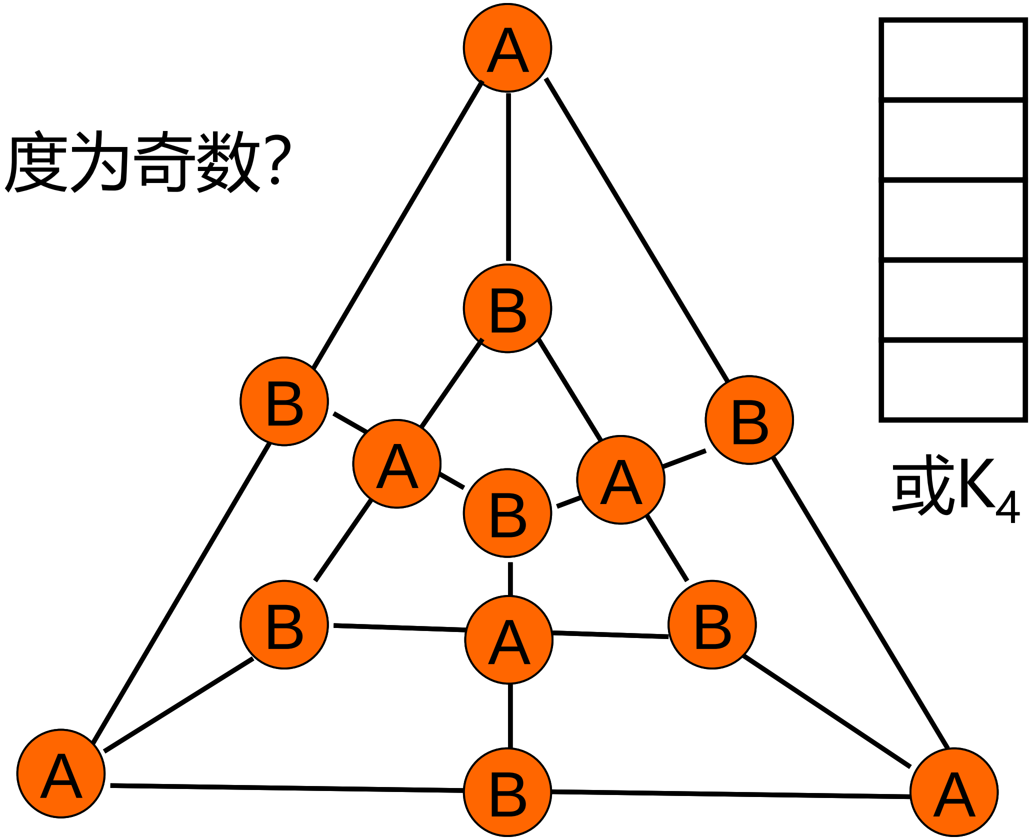
哈密顿道路与回路(13)

- 回顾
  - 若简单图G中任两点u,v, 恒有d(u)+d(v) ≥n-1，则G中存在Hamilton道路
  - 若简单图G中任两点u,v, 恒有d(u)+d(v) ≥n，则G中存在Hamilton回路
  - 若简单图G中任意一点v, 有d(v) ≥n/2，则G中存在Hamilton回路
- 推论
  - 若简单图G(n>2)的闭合图是完全图，则G有Hamilton回路

哈密顿道路与回路(14)

- 例（脑筋急转弯）
  - 证明下图没有H回路

度为奇数?

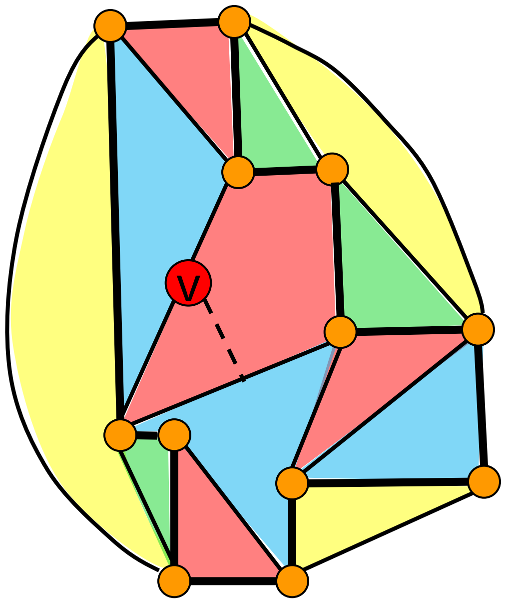


或K_4

- 若给某个结点标记A，其相邻结点标记B，B的邻结点标为A，恰好把图标完
- 若G中有H回路，则必是ABAB…AB的形式，但由于A与B数目不同，所以不存在H回路

哈密顿道路与回路(15)

- 例：4色猜想……
  - 若一个地图中有H回路，则可用4种不同颜色对域进行着色，使相邻域(共边)颜色不同。
    - 找到图中一个H回路（粗线表示）
    - H将图分成回路内外两部分
    - 每部分内都不存在三个(以上)区域互相相邻的情形
      - 否则会存在如图中v这样的点！
      - 存在这样的点与H是哈密顿回路矛盾
    - 因此回路内或外都只用两种颜色可以区分



哈密顿道路与回路(16)

- 哈密顿回路与欧拉回路

回路名称	欧拉回路	哈密顿回路
回路类型?	简单回路	初级回路
回路定义?	过所有边	过所有点
如何判断?	有充要条件	无直接判断的 充要条件

哈密顿道路与回路(17)

- 哈密顿回路与欧拉回路

回路名称	欧拉回路	哈密顿回路
回路类型?	简单回路	初级回路
回路定义?	过所有边	过所有点
如何判断?	有充要条件	无直接判断的 充要条件

道路存在性之后呢?　　唯一性?最优回路?

旅行商问题与分支定界法

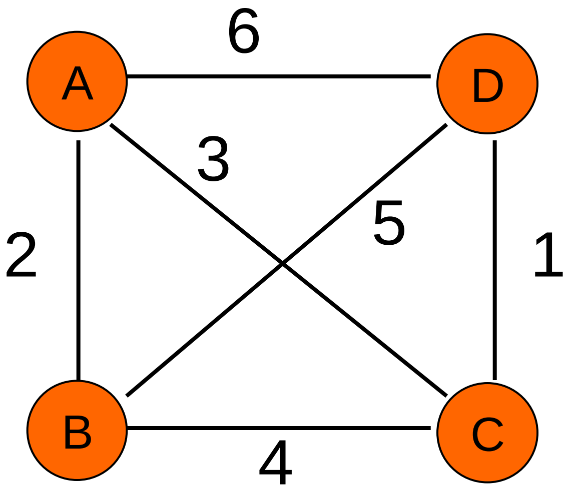
游览下述城市并返回，
如何规划路线?

北京、广州、天津、厦门、
青岛、深圳、大连、三亚

- 旅行商问题
  - 给定一个正权完全图，求其长最短的H 回路

各就各位：AI开找……

最短的H回路是(A,B,D,C,A)，长为11



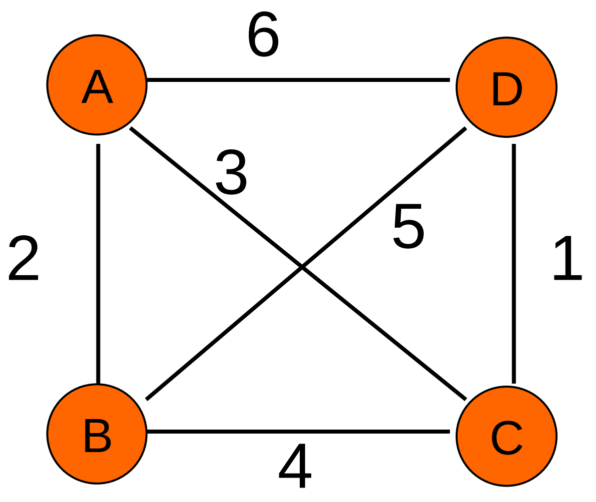
旅行商问题与分支定界法(2)

- 求解旅行商问题

算法之前来点基础理论……

  - 枚举法
    - n个结点的完全图有多少个不同的H回路?

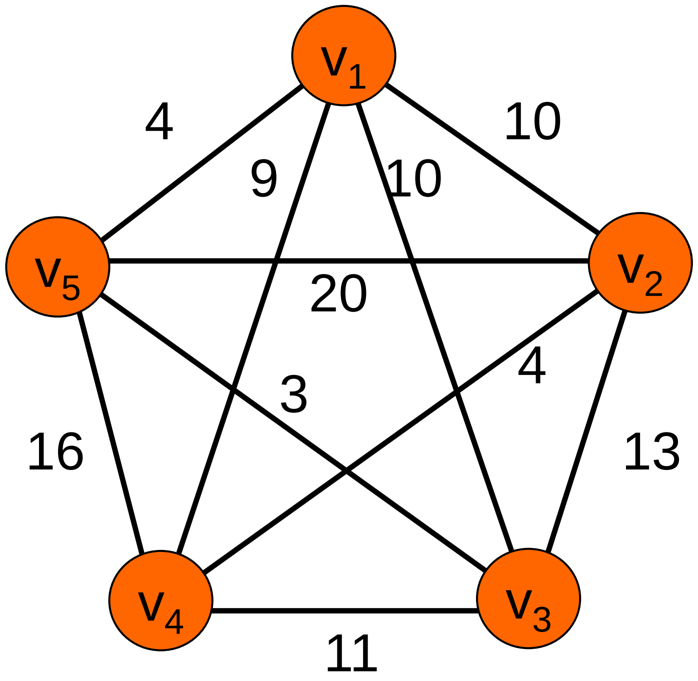
$$\frac{1}{2}(n-1)!$$



    - 完全穷举复杂度太大
  - 求解精确解的最佳方法
    - 分支定界法
    - 考虑最短的H回路是(A,B,D,C,A)，长为11?
    - 先选较短边，探测次短边，利用现有“最佳”路径的结果!

旅行商问题与分支定界法(3)

- 例：求图的最短H回路
  - 使用分支定界法
  - 将边权排序后可得

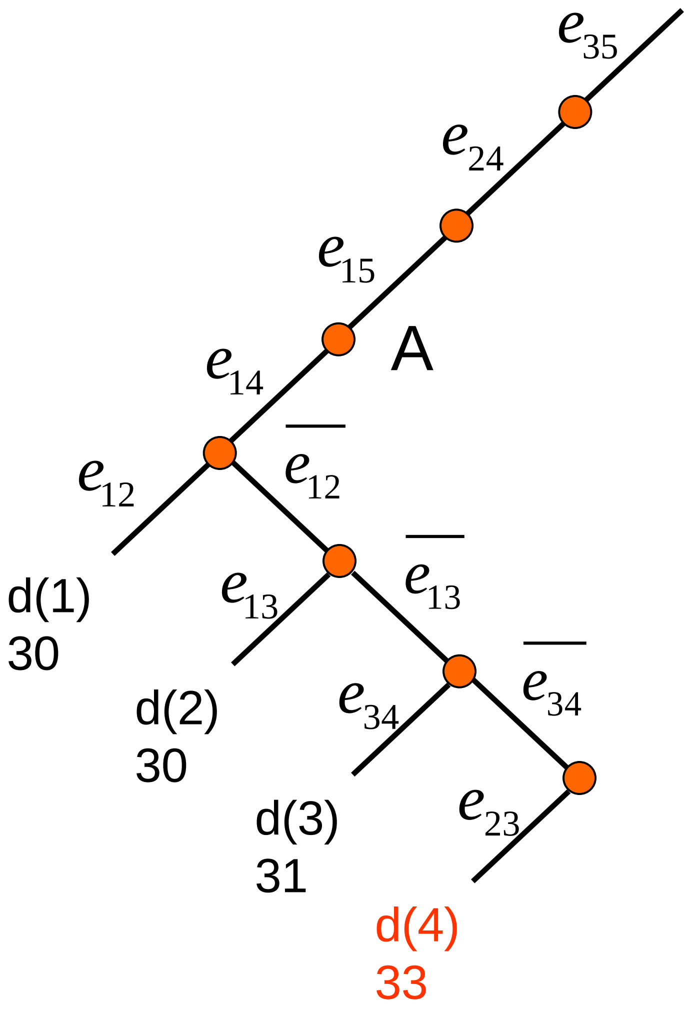


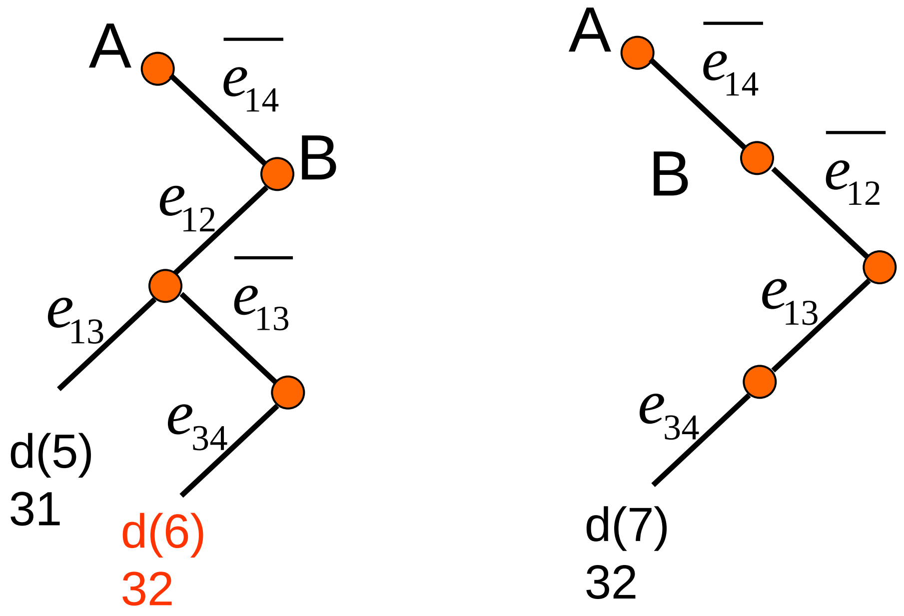
$$\begin{array}{c} l_{ij} \\ e_{ij} \end{array} \begin{pmatrix} 3 & 4 & 4 & 9 & 10 & 10 & 11 & 13 & 16 & 20 \\ e_{35} & e_{24} & e_{15} & e_{14} & e_{12} & e_{13} & e_{34} & e_{23} & e_{45} & e_{25} \end{pmatrix}$$

  - 使用深探法构造分支

旅行商问题与分支定界法(4)

$$\begin{array}{c} l_{ij} \\ e_{ij} \end{array} \begin{pmatrix} 3 & 4 & 4 & 9 & 10 & 10 & 11 & 13 & 16 & 20 \\ e_{35} & e_{24} & e_{15} & e_{14} & e_{12} & e_{13} & e_{34} & e_{23} & e_{45} & e_{25} \end{pmatrix}$$

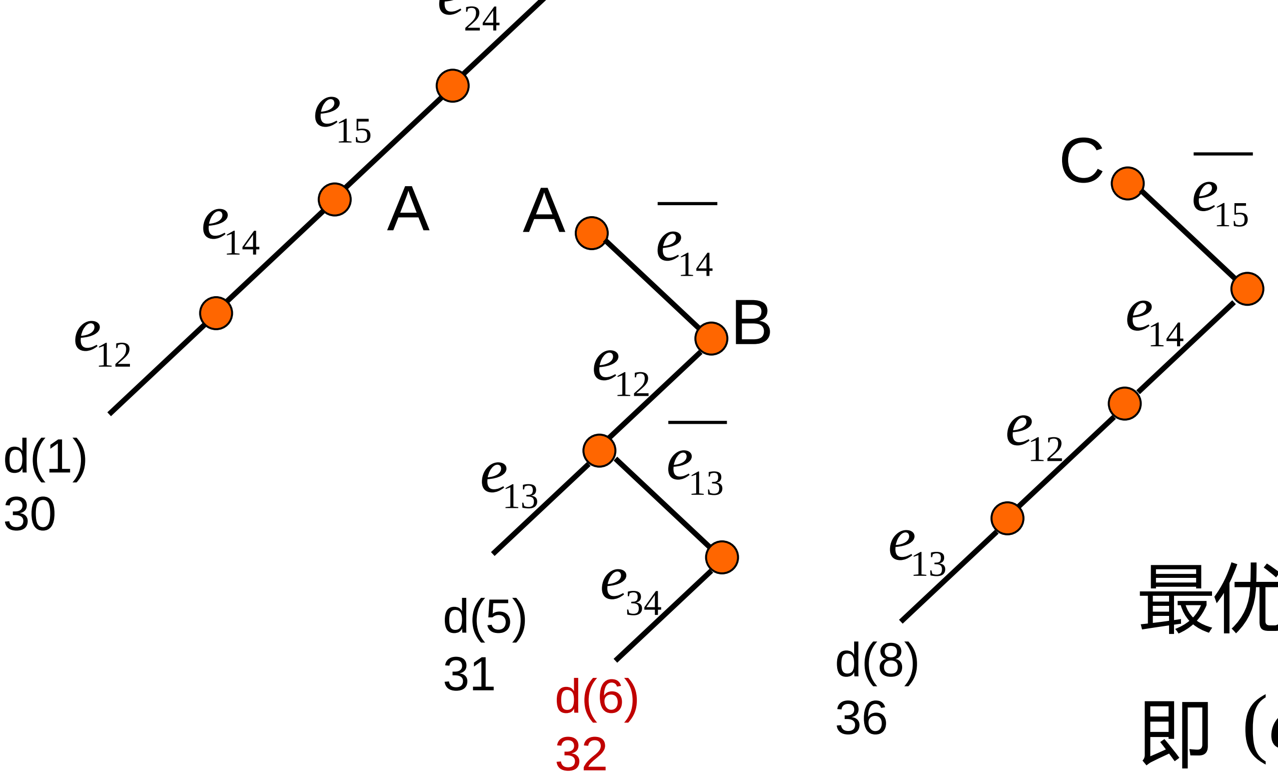




旅行商问题与分支定界法(5)

d(6) = 32

$$
\begin{array}{c}
l_{ij} \\
e_{ij}
\end{array}
\begin{pmatrix}
3 & 4 & 4 & 9 & 10 & 10 & 11 & 13 & 16 & 20 \\
e_{35} & e_{24} & e_{15} & e_{14} & e_{12} & e_{13} & e_{34} & e_{23} & e_{45} & e_{25}
\end{pmatrix}
$$



最优解是d(6)=32

即 $(e_{35}, e_{51}, e_{12}, e_{24}, e_{43})$

旅行商问题与分支定界法(3)

- 分支定界法

1. 将权由小到大排序，初始界为d_0足够大。

$$\begin{array}{c} l_{ij} \\ e_{ij} \end{array}\begin{pmatrix} 3 & 4 & 4 & 9 & 10 & 10 & 11 & 13 & 16 & 20 \\ e_{35} & e_{24} & e_{15} & e_{14} & e_{12} & e_{13} & e_{34} & e_{23} & e_{45} & e_{25} \end{pmatrix}$$

2. 在边权序列中依次选边进行深探，直到选取n条边，记为s，判断是否构成H回路
   - 每个结点标号只出现两次
   - 若构成H回路，用d(s)替换d_0，结束。
3. 若尚未构成H回路：继续深探
   - 依次删除当前s中最长的边，加入后面第一条待选边，进行深探。若它是H回路，且d(s)< d_0，则用d(s)替换d_0 ，转4;否则转3
4. 退栈过程
   - 不能再深探或d(s)>= d_0时，需要退栈
   - 若栈空则结束，最佳值为d_0 ;否则，如果新分支的d(s)>= d_0，继续退栈；若d(s)< d_0 则转3

旅行商问题与分支定界法(6)

- 分支定界法特点

$$\begin{array}{c} l_{ij} \\ e_{ij} \end{array}\begin{pmatrix} 3 & 4 & 4 & 9 & 10 & 10 & 11 & 13 & 16 & 20 \\ e_{35} & e_{24} & e_{15} & e_{14} & e_{12} & e_{13} & e_{34} & e_{23} & e_{45} & e_{25} \end{pmatrix}$$

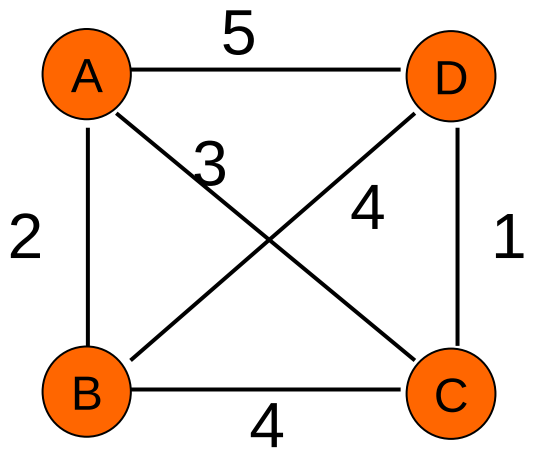
  - 搜索过程实质
    - 不断构造分支与确定新界值
    - 不搜索大于界值的分支
    - 最后得到的界值是最佳解吗?

搜索多少次?

  - 复杂度
    - 由于使用“剪枝”，该法显然比枚举法优越
    - 最坏情况下（不断尝试），复杂度仍为O(n!)
    - 怎么办呢?？？

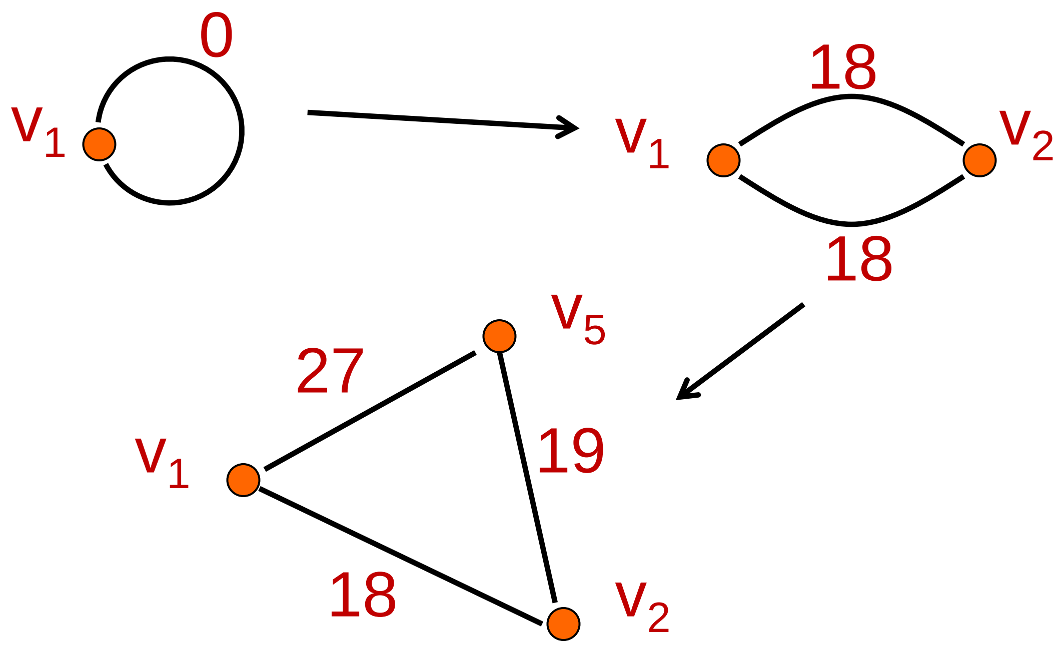
本质?连问三个为什么!

旅行商问题与分支定界法



$$\begin{bmatrix} 0 & 18 & 35 & 25 & 27 \\ 18 & 0 & 23 & 21 & 19 \\ 35 & 23 & 0 & 17 & 28 \\ 25 & 21 & 17 & 0 & 24 \\ 27 & 19 & 28 & 24 & 0 \end{bmatrix}$$

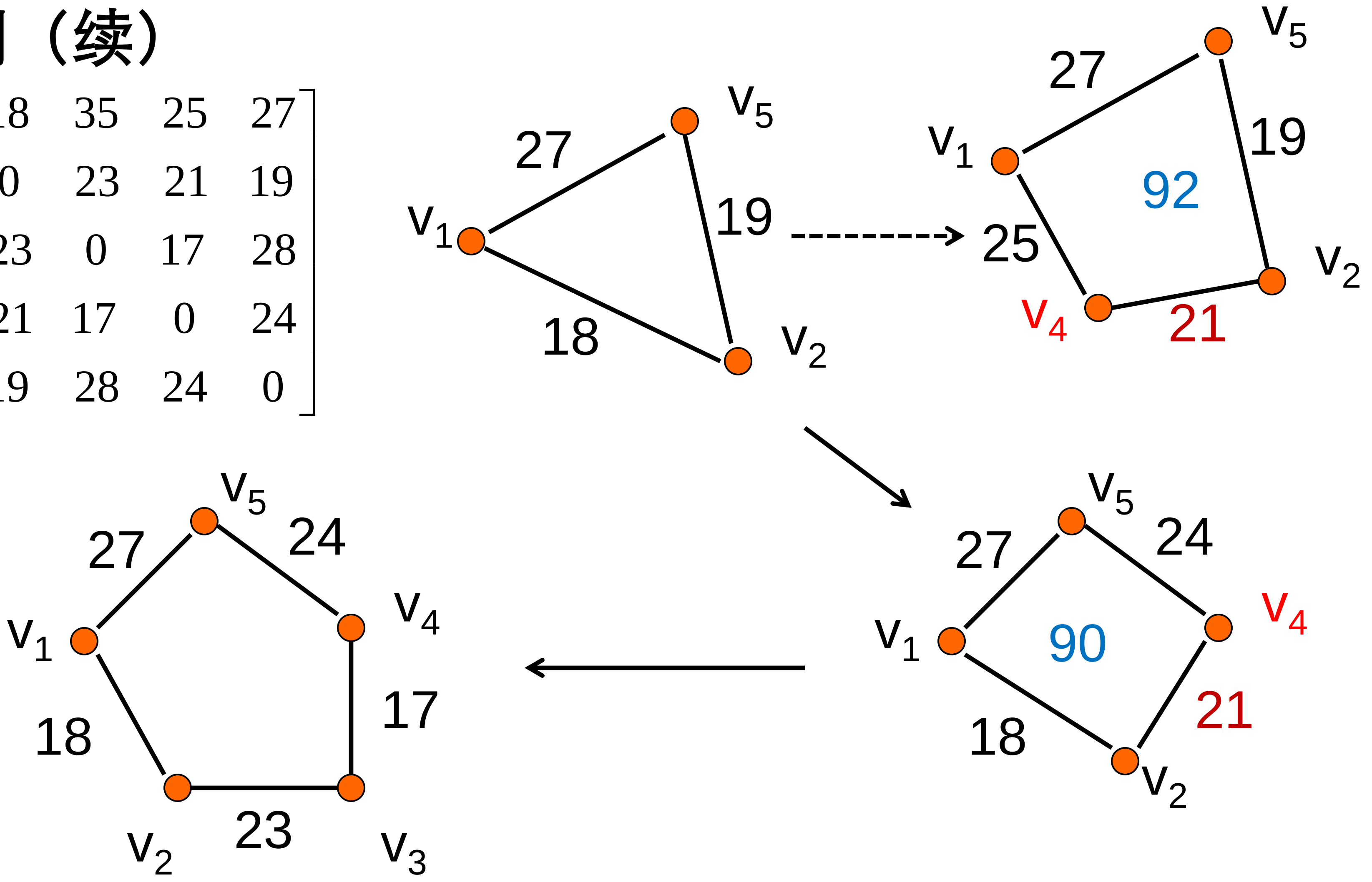
- 例：求旅行商问题近似解：“便宜”算法
  - 已知图G的权矩阵，使用便宜算法求解其旅行商问题的近似解



旅行商问题与分支定界法

- 例（续）

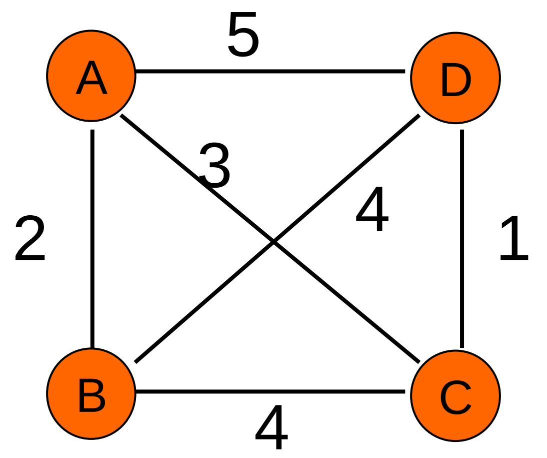
$$
\begin{bmatrix}
0 & 18 & 35 & 25 & 27 \\
18 & 0 & 23 & 21 & 19 \\
35 & 23 & 0 & 17 & 28 \\
25 & 21 & 17 & 0 & 24 \\
27 & 19 & 28 & 24 & 0
\end{bmatrix}
$$

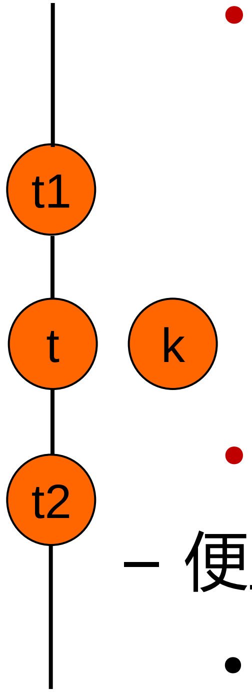


旅行商问题与分支定界法

- 求旅行商问题近似解——“便宜”算法
  - 构造不断扩充的初级回路T
    - 最初T是一个自环
    - 寻找与T最近的结点k，将k插入T
      - 设k与T中的t最近，具体插入t的位置需要依据k插入后回路T长度增量的大小而定
      - 若$w(k,t)+w(k,t_1)-w(t,t_1)<=w(k,t)+w(k,t_2)-w(t,t_2)$ 则k插到t与t_1之间; 否则k在t与t_2之间
    - 更新不在T上的节点到T的距离
  - 便宜算法的计算复杂度为$O(n^2)$
    - 每加入一个点O(n)（寻找n+更新n)），加入n个点

循环n次





旅行商问题与分支定界法

- 便宜算法是启发式算法
  - 启发式规则：加入当前最近的节点（距离当前回路圈）
  - 现在的决定是局部最优还是全局最优?
  - 将来还会修改现在的决定吗?
- 定理（便宜算法的性能）
  - 设正权完全图的边权满足三角不等式，其旅行商问题的最优值为Q，便宜算法的值是T，则$T/Q<2$。

发明定理：
近似性定理

旅行商问题与分支定界法

- 分支定界法与“便宜”算法
  - 质量（性能）
    - 从理论上讲“便宜”算法近似程度并不理想, 便宜值T与最优值Q相比只能保证T/Q<2
    - 实际中与最优解非常接近: 上例中，便宜算法的解是109，使用分支定界法是107
  - 效率（计算复杂度）
    - “便宜”算法大大优于分支定界法: $O(n^2)$ vs. $O(n!)$

实际生活的例子：游览下述城市并返回，如何规划路线（北京、广州、天津、厦门、青岛、深圳、上海）？

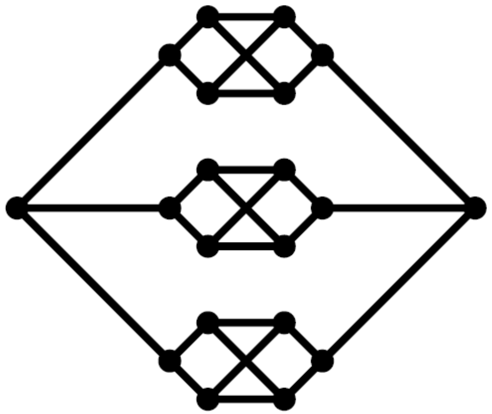
考虑哪些因素? 距离、列车时刻、停留时间……

总结：道路与回路进阶

- 欧拉道路与回路
  - 不要急于求解，而用大招：定义、定理、性质
  - 充要条件（构造法）
- 哈密顿道路与回路
  - 充分条件（极长）： $d(v)+d(u)\geq n-1$
  - 充要条件：闭合图☹ （存在唯一性）
  - 相邻人认识的例题：建模描述精准吗？
  - 初识四色猜想：妙用哈密顿回路
- 旅行商问题与分支定界法
  - 精确求解：分支定界法（透过现象看本质）
  - 近似求解：便宜算法（性能不仅仅是复杂度）

作业-欧拉图、H回路和旅行商

- 习题二（P53）
  - H回路: 第18，20，22题
  - 旅行商：分别用分支定界法和便宜法解第31题
- 补充题
  - （1）凸n边形及n-3条在形内不相交的对角线组成的图形称为一个剖分图。
    求证：当且仅当3|n时，存在一个剖分图是可以一笔画的圈。



  - （2）脑筋急转弯：右图是否有哈密顿回路？若有请指出；如无请证明。
  - （3）开放题：以欧拉与H回路为例，总结创新与写书思路（约300字）。

道路与回路的判定

- 代码实现BFS和DFS的思路
  - 存储待访问的顶点：队列和栈
    - C++中数组、vector、queue、stack 等都可以
  - 标记已访问的点
    - C++中 bool visited[N] 或是 set、map 等
- 更多参考
  - 语言的文档，如 https://zh.cppreference.com/
  - OI Wiki: https://oi-wiki.org/graph/dfs/
  - HackerEarth: https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/tutorial/

道路与回路的判定

```
BFS (G, s)
        queue Q                              存储待访问点的队列
        bool visited[ G.V.size() ]           标记顶点是否已访问
        Q.push( s )
        visited[ s ] = true                  从源 s 开始 BFS
        while ( Q not empty )
                v = Q.pop()
                // do something              对每个点 v 进行需要的操作
                for w in v.neighbours()
                        if ( not visited[ w ] )将 v 的所有未访问邻居入队
                                Q.push( w )
                                visited[ w ] = true
```

道路与回路的判定


```
DFS (G, s)
    stack S                        存储待访问点的栈（与 BFS 不同）
    bool visited[ G.V.size() ]     标记顶点是否已访问
    S.push( s )
    visited[ s ] = true            从源 s 开始 DFS
    while ( S not empty )
        v = S.pop()
        // do something            对每个点 v 进行需要的操作
        for w in v.neighbours()
            if ( not visited[ w ] )将 v 的所有未访问邻居入栈
                S.push( w )
                visited[ w ] = true
```