路与连通

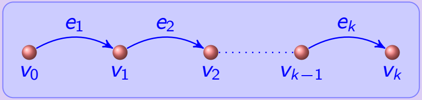
School of Computer
Wuhan University

1. 路径
   - 路径与基本路径
   - 连通性
   - 回路与基本回路
   - 距离与直径
2. 路径的应用
   - 赋权图与最短路径
   - Euler路径与回路
   - Hamilton路径与回路

1 路径
- 路径与基本路径
- 连通性
- 回路与基本回路
- 距离与直径

2 路径的应用
- 赋权图与最短路径
- Euler路径与回路
- Hamilton路径与回路

路径、简单路径和基本路径

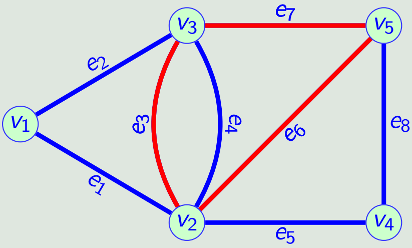


设$D=\langle V,E\rangle$(或G)是一图，$P=(v_1e_1v_2e_2v_3\cdots v_ke_kv_{k+1})$是一结点与边交错序列，且序列中相邻的点与边关联，称这样的序列为路径(链，通路，walk)；

更进一步：

- e_1的起点(端点)是路径的起点(端点)；e_k的终点(端点)是路径的终点(端点)，其余的点称为内部点.
- 序列中边的个数k称为路径的长度.
- 同一条边在P中仅出现一次，称为简单路径(trail，迹)；
- 同一结点在P中仅出现一次，称为基本路径(path，路)；
- P的起点点和终点相同称为回路(closed，闭链)；
- 简单回路(circuit，闭迹)，基本回路(圈)(cycle)；
- 对简单图P可简记为$(v_1v_2\cdots v_n)$或(v_1,v_n).

Example



- $(v_1 e_1 v_2 e_3 v_3 e_7 v_5 e_7 v_3 e_7 v_5 e_7 v_3)$ 是路径(有重复的边);
- $(v_1 e_2 v_3 e_3 v_2 e_4 v_3)$ 是简单路径(无复边, 有重复点);
- $(v_1 e_1 v_2 e_5 v_4 e_8 v_5 e_7 v_3)$ 是基本路径(无重复点), 也是简单路径(无重复边);
- $(v_2 e_3 v_3 e_4 v_2 e_3 v_3 e_4 v_2)$ 是回路(端点重合), 但不是简单(有重复边);
- $(v_2 e_3 v_3 e_4 v_2 e_6 v_5 e_8 v_4 e_5 v_2)$ 是简单回路(无重复边), 不是基本回路;
- $(v_2 e_3 v_3 e_7 v_5 e_6 v_2)$ 是基本回路(无重复点), 也是简单回路.

相关性质

Propositions

- 基本路径也是简单路径，基本回路也是简单回路；
- 基本路径的长度$\leqslant n-1$，简单路径的长度$\leqslant m$；
- 基本回路的长度$\leqslant n$，简单回路的长度$\leqslant m$.

Proof.

- 长度为k的路径经过$k+1$个结点；
- 抽屉原则.

□

Definition

- 长度最长的基本路径称为最长路径；
- 包含每个点的基本路径称为Hamilton路径. Hamilton路径的长度等于$n-1$.

路径与基本路径的关系

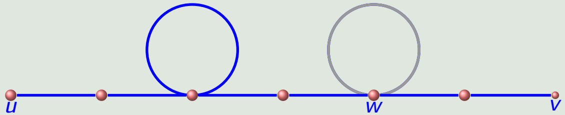
Proposition

若u到v有路径，则u到v也有基本路径.

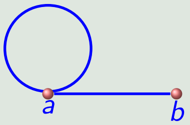
Proof(归纳法)

对u, v路径的长度l用归纳法:

- $l = 0$，即路径W无边，因此$W = (u)$是条基本路径.
- 设W的长度$< l$时结论成立，即若有长度$< l$的(u, v)路径，则有(u, v)基本路径.
- 设有长度为l的(u, v)路径，若该路径中没有重复结点，W即是一条(u, v)基本路径. 否则:
- 设l有重复结点w， 则从w点与边的序列中删除两次w出现中间的点与边后还是一条(u, v)路径，且该路径的长度比l要小. 根据归纳假设有(u, v)基本路径.



Example 1/2



求上图中从a到b长度为n的路径总数.

Solution[利用路径递归增长的属性].

设长度为n的路径总数为s_n, 则:

- $s_0 = 0$;
- $s_1 = 1$;
- $s_n = s_{n-1} + s_{n-2}$;
- $s_n = f_{n-1}$ (Fibonacci序列).

$\square$

Example 2/2

每个竞赛图都含有一个Hamilton有向路径.

Proof (归纳法)

设竞赛图的阶数为n, 则

- $n = 2$时结论成立;
- 设阶数为n的竞赛图T_n有一个Hamilton路径$v_1 v_2 \cdots v_n$;
- 设T_{n+1}是增加点v_{n+1}的竞赛图, 设$i \in \{1, 2, \ldots, n\}$且$i$是最大的使得$v_i v_{n+1}$是图$T_{n+1}$的边的下标.
- 如果$i = n$, 则$(v_1 v_2 \cdots v_n v_{n+1})$是图$T_{n+1}$的一条Hamilton路径.
- 如果$i < n$, 则$v_{i+1} v_{n+1} \notin E_{n+1}$, 这样$v_{n+1} v_{i+1} \in E_{n+1}$. 即$(v_1 v_2 \cdots v_i v_{n+1} v_{i+1} \cdots v_n)$是$T_{n+1}$的一条Hamilton路径.

$\square$

连通性

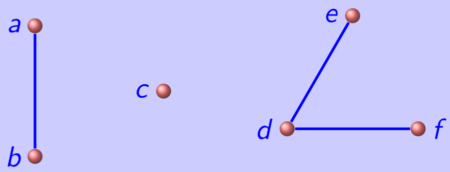
Definition

- 设$G(D)$是无向图(有向图), $u, v \in V$, 称u到v是可达的(accessible), iff, 从u到v有条有向路径(有向图); u和v之间有条路径(无向图);
- 无向图中如果任意两个结点都是可达的, 称为连通图(connected).

Proposition

- 在无向图G中定义关系$\mathcal{R} \subseteq V^2$, $u\mathcal{R}v$, iff, u和v是可达的; 则$\mathcal{R}$是V上的等价关系;
- 设$V/\mathcal{R} = \{ V_1, V_2, \ldots, V_p \}$, G_i是V_i导出的子图; 则:
  - $G = \bigcup_{i=1}^{p} G_i$;
  - $G_i \cap G_j = \emptyset$, if $i \neq j$;
  - G_i称为连通分支; $|V/\mathcal{R}|$称为连通分支数.

Example



$\{a, b\}$ $\{c\}$ $\{d, e, f\}$

G_1 G_2 G_3

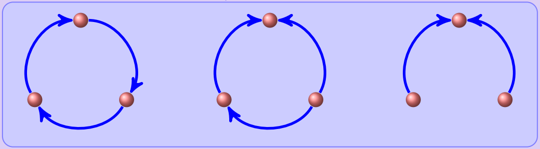
有向图的连通性

Definition

- 称有向图是强连通的(Strong connected), iff, 任意两个结点u和v, u到v是可达的, 并且v到u是可达的;
- 称有向图是单向连通的, iff, 任意两个结点u和v, u到v是可达的, 或者 v到u是可达的;
- 称有向图是弱连通的(Weakly connected), iff, G的底图是连通的.

Example

强连通 $\Longrightarrow$ 单向连通 $\Longrightarrow$ 弱连通

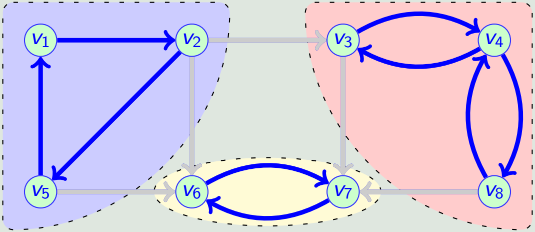


强连通分支

Proposition

- 定义关系$\mathcal{R} \subseteq V^2$, $u\mathcal{R}v$ iff u到v可达且v到u可达. 则R也是强连通图上的等价关系.
- 设$V/\mathcal{R} = \{ V_1, V_2, \ldots, V_p \}$, D_i是V_i导出的子图, D_i称为强连通分支. 一般 $D \neq \bigcup_{i=1}^{p} D_i$;
- $|V/\mathcal{R}|$称为强连通分支数.

Example

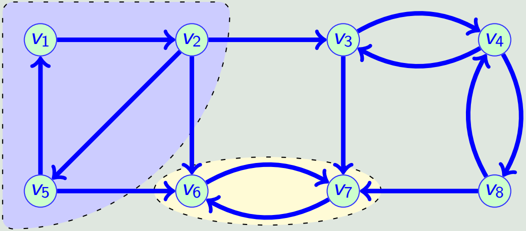


边集合

Definition

- D是有向图，S和T是V的两个不相交的非空子集合，定义 $(S,T)=\{e\,|\,e\in E\wedge\psi(e)=\langle s,t\rangle\wedge s\in S\wedge t\in T\}$;
- $[S,T]=(S,T)\cup(T,S)$;
- G是无向图时，$[S,T]$定义为两个端点分别落在S和T的边的集合.

Example



- 设$S=\{v_1,v_2,v_5\}$，$T=\{v_6,v_7\}$.
- $(S,T)=\{v_2v_6,v_5v_6\}$. $(T,S)=\varnothing$, $[S,T]=(S,T)$.

连通图的一个充要条件

Theorem

1. G是连通图iff对V的任意非空真子集S都有$[S,\overline{S}]\neq\varnothing$
2. D是强连通图iff对V的任意非空真子集S都有$[S,\overline{S}]\neq\varnothing\wedge[\overline{S},S]\neq\varnothing$

Proof

$\Longrightarrow$ (反证法)

- 设存在V的真子集S，使得$[S,\overline{S}]=\varnothing$.
- 设$u\in S\wedge v\in\overline{S}$. 设$ue_1x_1e_2x_2\cdots x_ne_nv$是一条连接$u$和$v$的基本路径.
- 设$x_i=\max\{j\,|\,x_j\in S\}$. 则$x_{i+1}\in\overline{S}\wedge\psi(e_{i+1})=x_ix_{i+1}$.
- 即$e_{i+1}\in[S,\overline{S}]\neq\varnothing$. 矛盾.

$\Longleftarrow$ (反证法)

- 设G不连通，则G至少有两个连通分支G_1和G_2, $G=G_1(V_1,E_1)\cup G_2(V_2,E_2)$.
- 则$V_2=\overline{V_1}$. $[V_1,\overline{V_1}]=\varnothing$. 矛盾.

Example 1

有向图D是强连通的, iff, 存在一条经过每个结点的回路.

Proof.

$\Longrightarrow$ 设$V = \{ v_1, v_2, \ldots, v_n \}$ 则:
$\exists (v_1 \ldots v_2) = P_1, (v_2 \ldots v_3) = P_3, \ldots, (v_n \ldots v_1) = P_n$;
$\therefore P_1 P_2 \cdots P_n$是条经过每个结点的回路;

$\Longleftarrow$ 设$P = (u_1 e_1 u_2 \ldots u_n e_n u_1)$是一条经过每个结点的回路,
$\forall v_i \neq v_j$, 则在P中存在$u_p = v_i \wedge u_q = v_j$, 设$p \leqslant q$,则:
$P' = (u_p \ldots u_q)$是v_i到v_j的路径;
$P'' = (u_q \ldots u_n e_n u_1 \ldots u_p)$是$v_j$到$v_i$的路径;
$\therefore D$是强连通的.

$\square$

Example 2 (归纳法)

设G是简单无向图，其中$n \geqslant 3$，如果m满足下述条件，则G是连通图：

$$m > \frac{(n-1)(n-2)}{2} = \binom{n-1}{2}$$

Proof(归纳法)

- 设$n = n \wedge m = m$.
- G没有孤立点，否则，设v是孤立点，则$G - \{v\}$是$(n-1, m)$图，而$n-1$个结点的简单无向图的最多边数是$\binom{n-1}{2} < m$，矛盾.
- 用归纳法证明：
  1. $n = 3$时，则$m > 1$，一定连通；
  2. 设$n = k - 1$时，G是连通图；
  3. 则$n = k$时，设$v \in V$，则$\deg(v) \geqslant 1$，否则是孤立点.
     — if $\deg(v) = n - 1$；则v和其他$n-1$个结点相邻，所以G连通.
     — if $1 \leqslant \deg(v) \leqslant n-2$，则$G' = G - \{v\}$，
     其中：$n(G') = n - 1$，$m(G') = m - \deg(v)$.
     $\therefore m(G') = m - \deg(v) > \binom{n-1}{2} - (n-2) = \binom{n-2}{2}$，
     由归纳假设G'是连通，$\therefore G$是连通的.

Example 2 (2)

设$G(V,E)$是简单无向图，其中$n \geqslant 3$，如果m满足下述条件，则G是连通图：

$$m > \frac{(n-1)(n-2)}{2} = \binom{n-1}{2}$$

Proof(反证法)

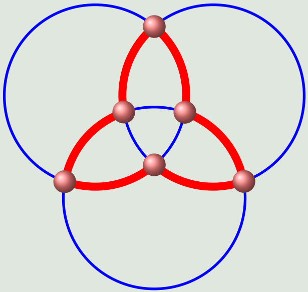
- 设G不连通，则$G = G_1(V_1,E_1) \cup G_2(V_2,E_2)$.

回路与基本回路

Definition

- C_n表示长度为n的基本回路.
- 长度为3的基本回路称为三角形.
- 长度最小基本回路的长度称为围长(girth), 记为$g(D)$.

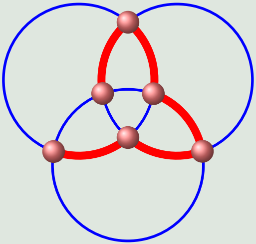
八面体对任意的$k(3 \leqslant k < n)$都有长度k的基本回路



Example

设$G(V,E)$是简单无向图，若$\delta \geqslant 2$. 则G含有一个长度至少为$\delta+1$的基本回路.

$\delta=4$有长度为5的基本回路



Proof(extremality)

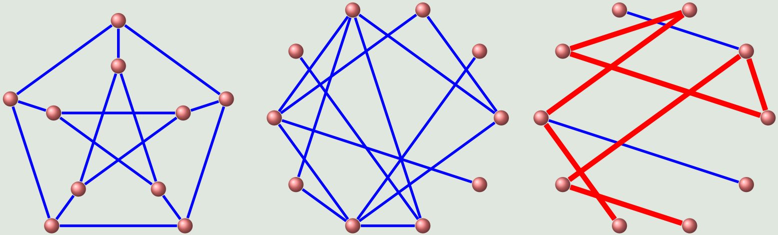
- 设$P=(v_0v_1\cdots v_k)$是G中长度最长的基本路径.
- 则对任意的与v_0相邻的点v都在路径P上，否则$(vv_0\cdots v_k)$是一个长度更长的基本路径.
- $|P| \geqslant \deg(v) \geqslant \delta$.
- 设v_l是P中最后一个与v_0相邻的点.
  则$C=(v_0v_1\cdots v_lv_0)$是一个长度不小于$\delta+1$的基本回路.

距离与直径

Definition

- 设$D(G)$是有向图(无向图), $u, v \in V$, $D(G)$中所有(u, v)基本路径中的最短长度称为从u到v的**距离**, 记为$d(u, v)$. 如果u到v不可达, 则约定$d(u, v) = \infty$
- **直径**: $d(D) = \max\{\, d(u, v) \mid u, v \in V \,\}$.
- $d(D) = 1 \iff K_n^* \subseteq D$, $d(G) = 1 \iff K_n \subseteq D$.

Example

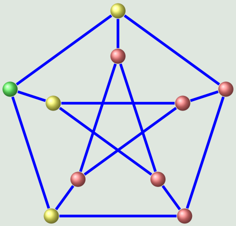


(a) 直径为2　　(b) 直径为4　　(c) 直径为7

Example

围长为5的k正则无向图至少有k^2+1阶.

3正则, 5围长



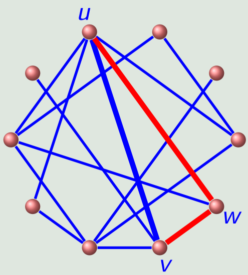
Proof

- 设$v \in V$, 定义$S_{v,i}$为与v距离为i的结点集合($i = 0, 1, 2$).
- 则$|S_{v,1}| = k$, 且$S_{v,1}$中的点互不相邻. 否则有三角形, 与围长5矛盾.
- 设$v', v'' \in S_{v,1}$. 则 $(S_{v',1} - \{v\}) \cap (S_{v'',1} - \{v\}) = \varnothing$. 否则有长度为4的基本回路.
- 而$S_{v',1} - \{v\} \in S_{v,2}$且$|S_{v',1} - \{v\}| = k - 1$.
- 因此$|S_{v,0} \cup S_{v,1} \cup S_{v,2}| = 1 + k + k(k-1) = k^2 + 1$.

Example

设G是简单图. 如果$d(G) > 3$, 则$d(\overline{G}) < 3$.

直径为4



Proof

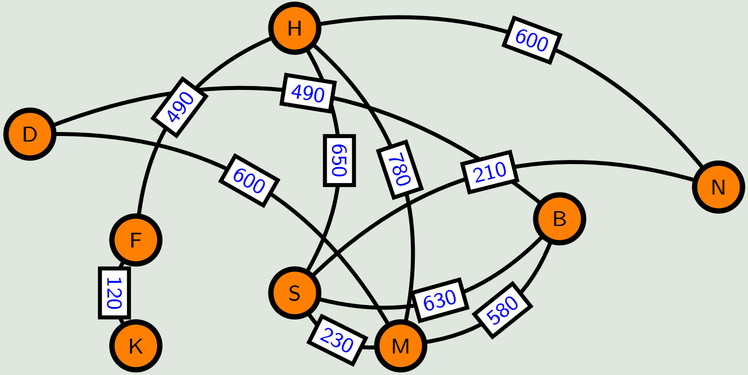
- 设$u, v \in V$, 若$uv \notin E(G)$, 则$uv \in E(\overline{G})$. $d_{\overline{G}}(u, v) = 1$.
- 若$uv \in E(G)$, 若在G中任意的一个结点至少与u和v其中之一相邻, 则$\forall x, y \in V$, $d_G(x, y) \leqslant 3$, 矛盾.
- 设$w \in V$, $uw, vw \notin V(G)$, 则$uw, vw \in V(\overline{G})$, 即$d_{\overline{G}}(u, v) \leqslant 2$.

1 路径
- 路径与基本路径
- 连通性
- 回路与基本回路
- 距离与直径

2 路径的应用
- 赋权图与最短路径
- Euler路径与回路
- Hamilton路径与回路

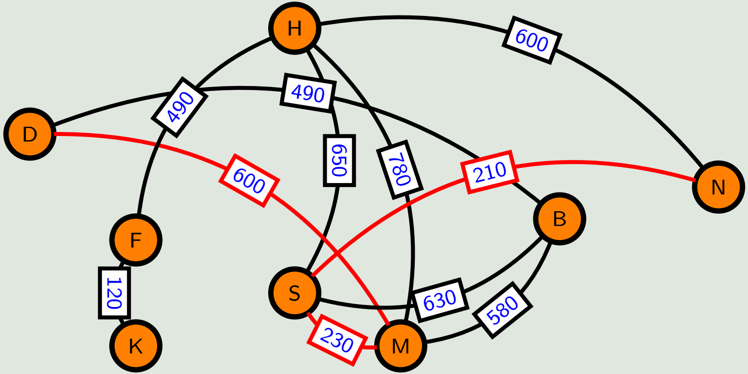
Example

8个城市的航线图，求城市D到城市N的最短航程



Example

8个城市的航线图，求城市D到城市N的最短航程



1. $P_1 = (DMHN)$，航程$600 + 780 + 600 = 1980$；
2. $P_2 = (DBSN)$，航程$490 + 630 + 210 = 1330$；
3. $P_3 = (DMSN)$，航程$600 + 230 + 210 = 1040$.

赋权图与最短路径

Definition

赋权图$G=\langle V,E,\varphi,\psi\rangle$，其中$\psi:E\longrightarrow\mathbb{R}^+$，设$P=(e_1,e_2,\ldots,e_k)$；则路径$P$的权值$W(P)$定义为：

$$W(P)\triangleq\sum_{i=1}^{k}\psi(e_i)$$

u到v的最短路径$d(u,v)$定义为：

$$d(u,v)\triangleq\begin{cases}\min\{\,W(P)\mid P\text{是u到v的路径}\,\};\\ \infty,\ \text{if u到v是不可达的}.\end{cases}$$

Remark

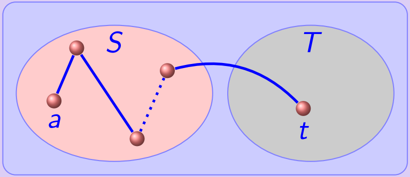
- 要求每个边的权值非负；
- Dijkstra's algorithm — 求解某个结点到其他结点的最短距离；
- Floyd-Warshall algorithm — 求解每对结点的最短距离.

相关定义

Definition

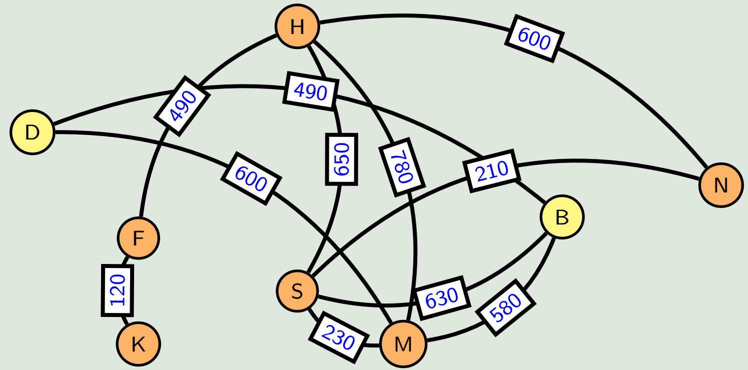
设$a \in V$，求a到其他结点的最短路径，将V划分为两个不相交的子集合S和T，其中$a \in S$；$t \in T$，定义：

$$D^T(t) = \text{从}a\text{到}t\text{不经过}T\text{中除}t\text{以外其他结点的路径的最小值}$$



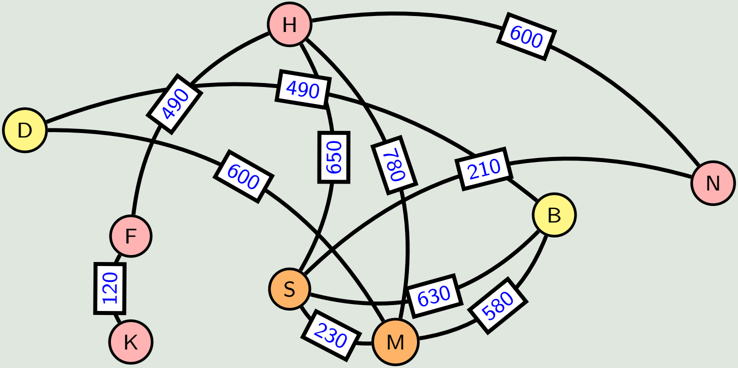
Example

$S = \{D, B\}$



Example

$S = \{ D, B \}$



1. $D_S(\mathrm{M}) = \min\{D(\mathrm{DBM}), D(\mathrm{DM})\} = 600$;
2. $D_S(\mathrm{S}) = D(\mathrm{DBS}) = 490 + 630 = 1120$;
3. $D_S(\mathrm{H}) = D_S(\mathrm{F}) = D_S(\mathrm{K}) = D_S(\mathrm{N}) = \infty$.

相关引理

Proposition

设$D^T(t_0) \triangleq \min\{\, D^T(t) \mid t \in T \,\}$, 则:

$$d(a, t_0) = D^T(t_0)$$

if 结点a到结点$v \in V - T$的最短路径不经过T中的结点.

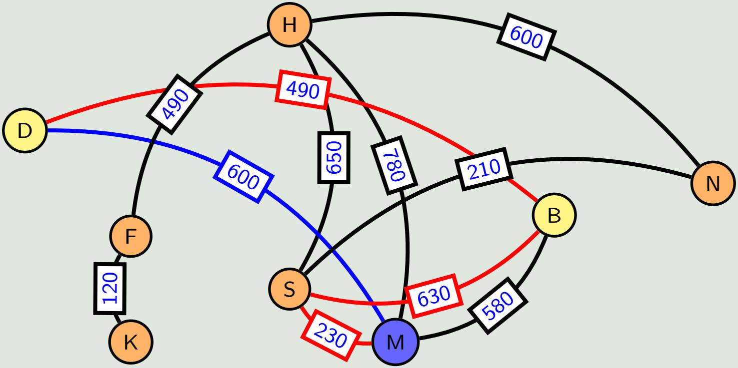
Proof(反证法).

- 设a到t_0最短路径为$P = (a, s_1, \ldots s_p, t, \ldots, t_0)$, 其中$t$为第一次出现的$T$中的结点, 即$P$还要经过$T$中的其他的结点;
- 设$P_1 = (a, s_1, \ldots s_p, t)$, $P_2 = (t, \ldots, t_0)$, 则$P = P_1 \cup P_2$; 并且P_1是a到t不经过T中除t以外其他结点的路径;
  $\therefore W(P_1) \geqslant D^T(t) \geqslant D^T(t_0)$;
- $\therefore W(P) = W(P_1) + W(P_2) > D^T(t_0)$, 与$W(P)$是最短路径矛盾.

□

Example: D(M)一定是D到M的最短路径

$S = \{\mathrm{D}, \mathrm{B}\}$



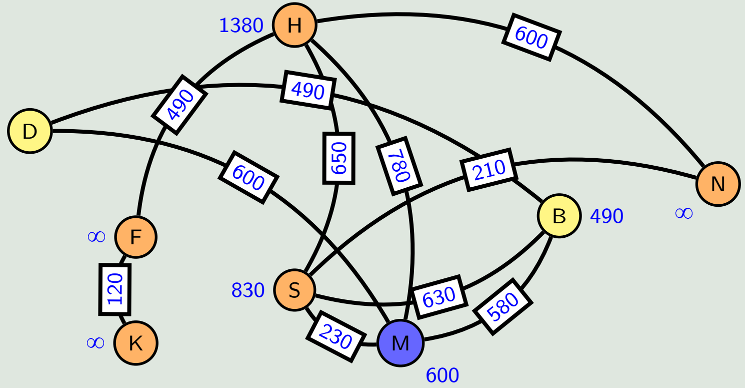
1. $D(\mathrm{DBM}) > D_S(\mathrm{M}) = \min\{D(\mathrm{DBM}), D(\mathrm{DM})\} = 600$;
2. $D(\mathrm{DBSM}) = D_S(\mathrm{S}) + 230 > D_S(\mathrm{M})$.

Dijkstra's Algorithm

Dijkstra's Algorithm

1. 初始化: $S := \{ a \}$, $T = V - \{ a \}$, $D^T(t) = \psi(a, t)$, $d(a, t) = \infty$;
2. 求$D^T(t_0) = \min\{ D^T(t) \mid t \in T \}$;
   $S := S \cup \{ t_0 \}$, $T := T - \{ t_0 \}$, $d(a, t_0) := D^T(t_0)$;
3. if $(T == \varnothing)$ then stop; else goto ②.

Example



1. $S = \{\mathrm{D}\}$, $D_S(\mathrm{B}) = \min\{ D_S(v) \}$;
2. $S = \{\mathrm{D}, \mathrm{B}\}$, $D_S(\mathrm{M}) = \min\{ D_S(v) \}$;
3. $S = \{\mathrm{D}, \mathrm{B}, \mathrm{M}\}$, $D_S(\mathrm{S}) = \min\{ D_S(v) \}$;

相对距离的计算

Proposition

设$D^T(t_0) \triangleq \min\{\, D^T(t) \mid t \in T \,\}$，设$S' = S \cup \{\, t_0 \,\}$, $T' = T - \{\, t_0 \,\}$，则：

$$D^{T'}(t) = \min\{\, D^T(t), D(t_0) + W(t_0, t) \,\}$$

注意：从a到S中结点的最短路径经过的结点一定在S中.

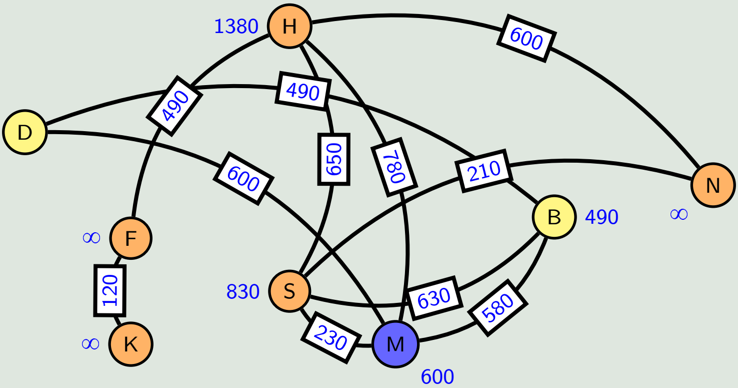
Proof.

设P是a到t不经过T'中除t以外其他结点的最短路径，则：

- P中不含有t_0, 则P也是a到t不经过T中除t以外其他结点的最短路径；$\therefore D^{T'}(t) = D^T(t)$;
- P中含有t_0:
  设$P = (a, s_1, s_2, \ldots, s_k, t_0, s'_1, s'_2, \ldots, s'_j, t)$，注意到$a$到$s'_j$的最短路径经过的结点仅在$S$中，$\therefore W(P) > d(a, s'_j) + W(s'_j, t)$, 这与$P$是$a$到$t$仅经过$S'$中的结点的最短路径矛盾；
  $\therefore P = (a, s_1, s_2, \ldots, s_k, t_0, t)$, 此时$W(P) = D(t_0) + W(t_0, t)$.

□

Example: 已知$D_{\{D,B\}}$，求$D_{\{D,B,M\}}$

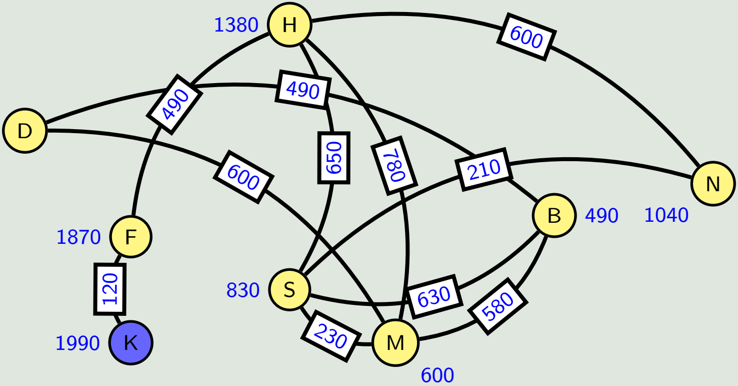


1. $D_{\{D,B,M\}}(S) = \min\{ D_{\{D,B\}}(S), D_{\{D,B\}}(M) + d(M,D) \}$;
2. $D_{\{D,B,M\}}(H) = \min\{ D_{\{D,B\}}(H), D_{\{D,B\}}(M) + d(M,H) \}$;
3. 其他结点与M不相邻，相对距离保持不变.

Dijkstra's Algorithm

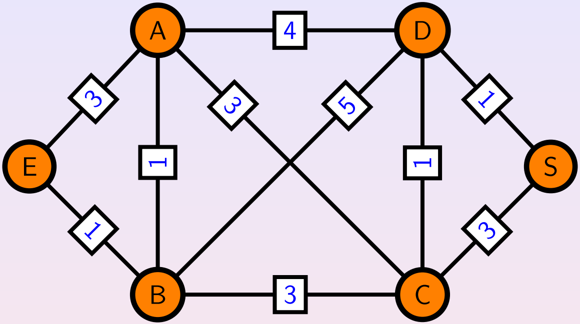
```
function Dijkstra(G, W, a)
  for each vertex v in V[G]  // Initialization
     do d[v] := w(a,v)
  t0 := undefined
  d[s] := 0
  T := set of all vertices
  while T is not an empty set
    do t0 := Extract-Min(T)
       T := T - { t0 }
       for each edge (t0, t) where t in T
         do if d[t] > d[t0] + w(t0,t) //Relax (u,v)
              then d[t] := d[t0] + w(t0,t)
```

Example



4. $S = \{\mathrm{D, B, M, S}\}$, $D_S(\mathrm{N}) = \min\{D_S(v)\}$;
5. $S = \{\mathrm{D, B, M, S, N}\}$, $D_S(\mathrm{H}) = \min\{D_S(v)\}$;
6. $S = \{\mathrm{D, B, M, S, N, H}\}$, $D_S(\mathrm{F}) = \min\{D_S(v)\}$;
7. $S = \{\mathrm{D, B, M, S, N, H, F}\}$, $D_S(\mathrm{K}) = \min\{D_S(v)\}$.

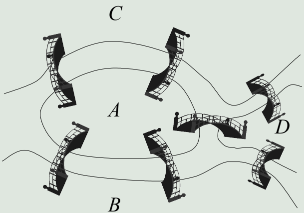
Dijkstra算法一个完整的例子

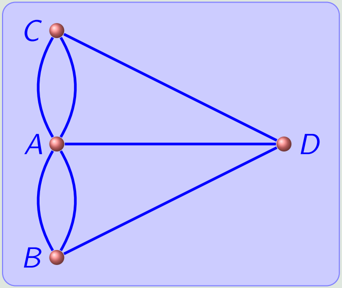


E	B	A	C	D	S	选择
0	∞	∞	∞	∞	∞	E
	1(E)	3(E)	∞	∞	∞	B
		2(B)	4(B)	6(B)	∞	A
			4(B)	6(B)	∞	C
				5(C)	7(C)	D
					6(D)	S

Euler路径与回路— Example

七桥问题(Königsberg Bridge)





Euler路径与回路

Definition

- 经过图中每个边一次并且仅一次的路径称为Euler路径;
- 经过图中每个边一次并且仅一次的回路称为Euler回路.

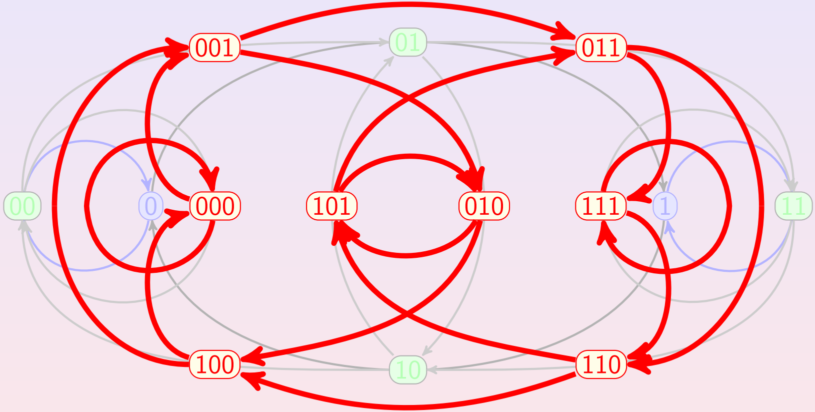
Theorem

- *D是Euler图 iff D是强连通的, 且每个结点的出度与入度相等.*
- *D有Euler路径 iff D是弱连通的，且最多有一个点的出度比入度大1，同时也最多有一个点入度比出度大1，其他的点出度和入度相等.*
- *G是Euler图 iff G中每个结点的度数均是偶数. G中仅有0个度数为奇数的结点.*
- *G有Euler路 iff G中仅有0个或2个度数为奇数的结点.*

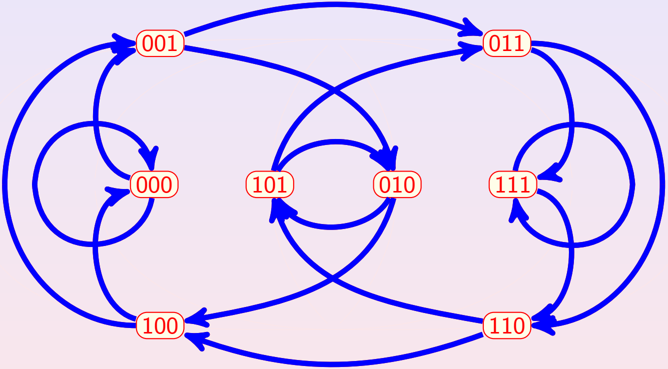
七桥问题

三个结点的度数为奇数，没有Euler回路，也没有Euler路径.

Example: De Brujin图：从$B(2,1)$到$B(2,3)$



Example: De Brujin图的任意Euler回所移出的数字串



Euler回:	000, 000, 001, 011, 111, 111, 110, 101, 011, 110, 100, 001, 010, 101, 010, 100, 000
移出数字:	0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1

De Brujin序列

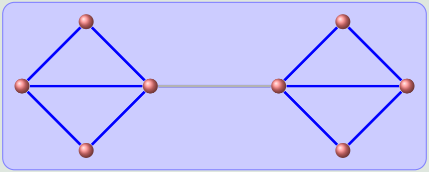
$$
\begin{array}{cccccccccccccccccc}
 & \{0 & 0 & 0 & 0\} & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & \{0 & 0 & 0 & 1\} & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & \{0 & 0 & 1 & 1\} & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & \{0 & 1 & 1 & 1\} & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & \{1 & 1 & 1 & 1\} & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & \{1 & 1 & 1 & 0\} & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & \{1 & 1 & 0 & 1\} & 1 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & \{1 & 0 & 1 & 1\} & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \{0 & 1 & 1 & 0\} & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & \{1 & 1 & 0 & 0\} & 1 & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \{1 & 0 & 0 & 1\} & 0 & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & \{0 & 0 & 1 & 0\} & 1 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & \{0 & 1 & 0 & 1\} & \\
 & 0\} & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \{1 & 0 & 1 & \\
\ldots & 0 & 0\} & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \{0 & 1 & \ldots \\
\ldots & 0 & 0 & 0\} & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & \{1 & \ldots
\end{array}
$$

割边

Definition

设G是无向连通图，$e \in E$称为割边(桥)，iff，G删除边e之后将增加连通分支数.

Example



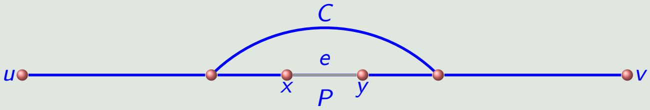
割边与圈的关系

Theorem

边e是图G的割边当且仅当边e不在任何基本回路上.

Proof(反证法)

1. 设H是包含边e(端点为x和y)的连通分支，则仅需证明结论对H成立.
2. 必要性：设e是割边，但是e在H的某个基本回路C上，需证明删除e后还保持连通，从而与割边的定义矛盾.
3. 设$u, v \in V(H)$. 若(u, v)路径不经过e，则u和v在$H - e$还是可达.
4. 若(u, v)路径经过e，可设 (u, v) 路径为 $(ue_1u_1 \cdots xeye_1'v_1 \cdots v)$，则 u 和 x，y 和 v 在图 $H - e$ 中可达，而 e 在回路 C 上，$\therefore$删除 e 后，x 和 y 还是可达，根据可达关系的传递性u和v在$H - e$中可达，即$H - e$是连通图. 矛盾.
5. 充分性：设e不在任何回路上，但$H - e$后还保持连通，即不经过 e 的 (x, y) 路径 P 存在，这样 P 加 e 即构成回路，从而导致矛盾.



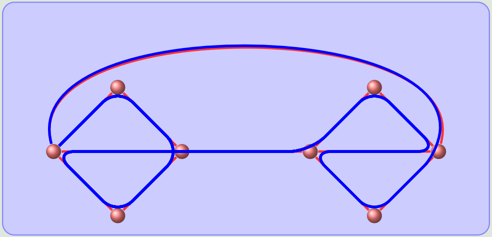
Fleury's Algorithm

Fleury's Algorithm

设$G(n.m)$是无向连通图，则：按下述算法构造出的边的序列一定是Euler回路；

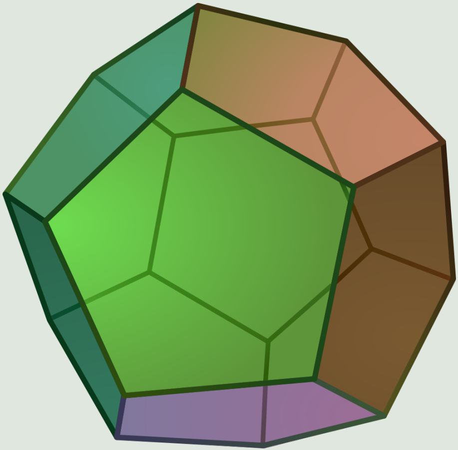
1. 初始化：$C := ()$，$v :=$ 任意一结点；
2. if $length(C) = m$, then Stop;
3. 设$C = (e_1, e_2, \ldots, e_k)$, if 图$G - \{ e_1, e_2, \ldots, e_k \}$有连接$v$一个非割边，则选择该非割边，否则选择连接$v$的一个割边，设所选的边为$e_{k+1}$；令$C := concate(C, e_{k+1})$；$v := e_{k+1}$的另一个端点；
4. goto ②.

Fleury's Algorithm动态演示



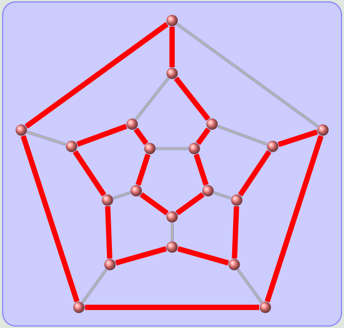
Hamilton路径与回路— Example (1/2)

Hamilton's " Round the World" Puzzle, 正12面体(Dodecahedron)



Hamilton路径与回路— Example(2/2)

Hamilton's " Round the World" Puzzle



Hamilton路径与回路

Definition

- 经过图中每个结点一次并且仅一次的路径称为Hamilton路径;
- 经过图中每个结点一次并且仅一次的回路称为Hamilton回路.

Theorem

设$G(n, m)$是简单无向图，则:

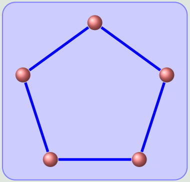
- *如果每对结点的度数之和大于$n-1$，则该图一定有条Hamilton路;*
- *如果每对结点的度数之和大于n，则该图一定有条Hamilton回路.*

Proof(连通性).

反证法：设G不是连通图，则$G = G_1 \cup G_2$，并且$G_1 \cap G_2 = \varnothing$，$|V_1| = n_1$，$|V_2| = n_2$，$n_1 + n_2 = n$，设$u \in V_1 \wedge v \in V_2$，则$\deg(u) \leqslant n_1 - 1 \wedge \deg(v) \leqslant n_2 - 1$，$\therefore \deg(u) + \deg(v) \leqslant n - 2$，矛盾. □

Counterexample

定理的条件不是必要条件的反例



- 有Hamilton回路;
- 但是每对结点的度数之和为$4 < 5 = n$.

Hamilton回路的一个必要条件

Theorem

设$G=\langle V,E\rangle$是Hamilton图(有Hamilton回路)，对任意的非空结点子集合$S\subseteq V$有:

$$\omega(G-S)\leqslant|S|$$

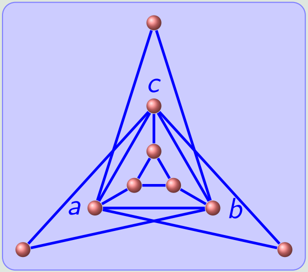
其中: $\omega(G-S)$是图$G-S$的连通分支数.

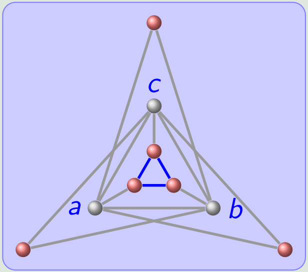
Proof.

设C是图G的一条Hamilton回路，从G中删除S中的结点后，最多将C分成$|S|$段路径，而$C-S$是$G-S$的子图，因此$G-S$的连通分支数不会大于$C-S$的连通分支数，故$\omega(G-S)\leqslant|S|$. $\square$

Example

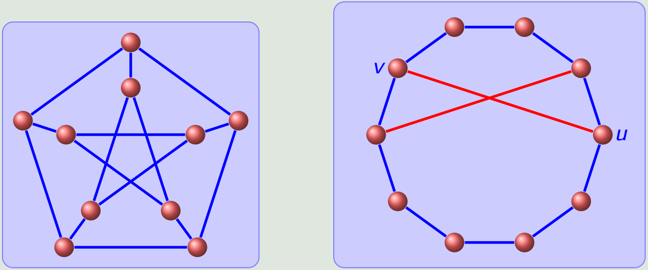
没有Hamilton回路





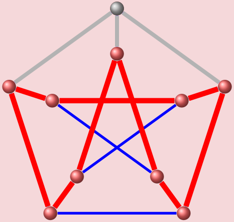
没有Hamilton回路　　删除a, b, c后连通分支数为$4 > 3$

反例：Petersen图不是Hamilton图(反证法)



- Petersen图的围长为5. 每个点的度数为3.
- 设有一个Hamilton回路，如右图所示.
- 则回路中点还要引出一边到回路上与该点不相邻的其他点. 且是在回路上与该结点的距离不小于4.
- 设每个结点引出的第三边与该结点的距离都是5，则有一个长度为4的基本回路，矛盾.
- 设某一点u的第三边连接在回路上与u距离为4的点v，则与u距离为5的点无论如何引边都产生长度为4的回路，矛盾.

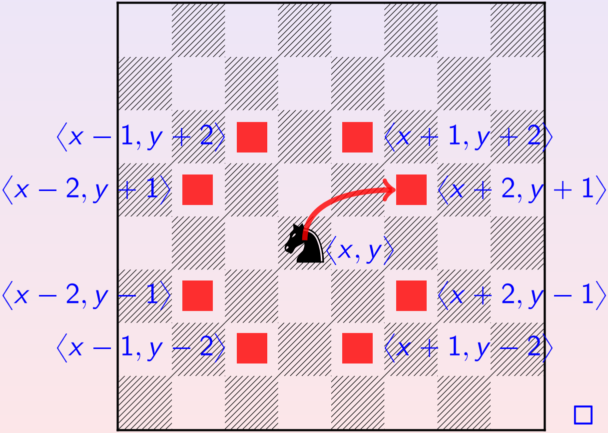
注释



- Petersen图有条Hamilton路径;
- Petersen图删除一个点后是Hamilton图;
- Petersen图的更多性质参见 http://en.wikipedia.org/wiki/Petersen_graph.
- 无向完全图K_n共有$\frac{(n-1)!}{2}$个不同的Hamilton回路.
- 有向完全图K_n^*共有$(n-1)!$个不同的Hamilton回路.
- 判断一个图是否Hamilton图是NP-complete problem. 见http://en.wikipedia.org/wiki/NP-complete.
- 只有暴力法(brute-force search)(也称穷举法(exhaustive search)).

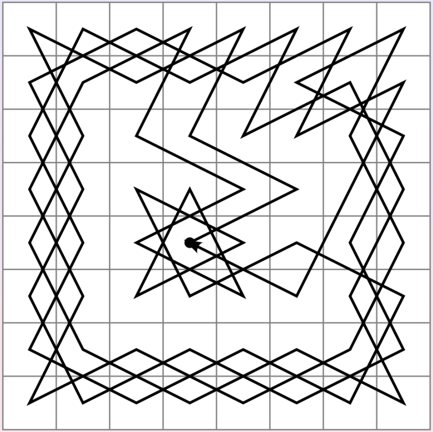
马跳棋盘问题(Knight tour)

国际象棋中马如何从棋盘中的某一格子出发没有重复地跳完棋盘的每一个格子(回到起点称为closed Knight tour).



□

马跳棋盘问题的一个解



A close Knight tour

搜寻马跳棋盘线路的暴力算法

- 初始化：

```
int b[8][8] = { 0 }; /* 棋盘，记录跳马的步数*/
int step = 1; /* 全局，记录当前的步数*/
int x = 0, y = 0; /* 当前马所在的位子，
                   可初始化任意位置*/
```

- 1. 如果x和y不在棋盘范围内，返回；
  2. 如果b[x][y]非零，表示该点已被访问，返回；
  3. b[x][y] = step++;
  4. 分别设置x和y为马可跳的下个点(共8个)，如x+2和y+1等，goto ①.
  5. 如果step <= 64，表示不成功，进行回溯，即将b[x][y]重新置零且step--.
- 如果step == 65，打印b，输出跳马线路.

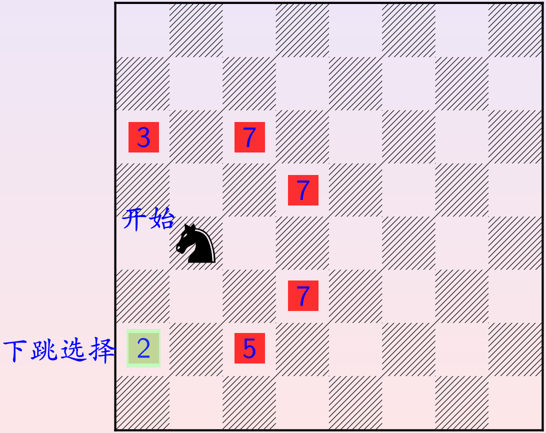
一个8×8的棋盘，如果要枚举出所有可能的路径，需要跳马4×10^{51}次！

源代码

```
int b[8][8] = {0}; /* see source file ktbrute.c */
int step = 1;
void traceKnight(int y, int x)
{
        /* 测试是否超出棋盘范围 */
    if (y < 0 || y > 8 - 1 || x < 0 || x > 8 - 1)  return;
        /* 测试该坐标已遍历 */
    if (board[y][x] != 0)  return;
    board[y][x] = step ++; /* 选择该坐标 */
      /* 递归 */
    traceKnight(y - 2, x - 1);
    traceKnight(y - 2, x + 1);
    traceKnight(y - 1, x + 2);
    traceKnight(y + 1, x + 2);
    traceKnight(y + 2, x + 1);
    traceKnight(y + 2, x - 1);
    traceKnight(y + 1, x - 2);
    traceKnight(y - 1, x - 2);
    if (step <= 8 * 8)   { /* 回溯 */
       board[y][x] = 0;
       step --;
    }
}
```

Warnsdorff 启发式算法(1823)

为了避免暴力算法中的非线性递归，算法在可能成为跳马下一跳的8格子中仅选一个，被选格是还有往下继续跳下去可能性最少的一个，即把还有更多可跳可能性的格子留给后续的跳马动作.



Warnsdorff算法

- 初始化：
```
    int b[8][8] = { 0 }; /* 棋盘，记录跳马的步数*/
    int x = 0, y = 0; /* 当前马所在的位子，
                          可初始化任意位置*/
```
- `for ( step = 2; step <=64; step ++ )`
    1. 设S是下一步可跳格子坐标所对应的集合；
    2. ∀ p ∈ S，求所有能成为p下一跳格子的数目N[p]；
    3. 设(x', y')是N[p]取最小对应的坐标；
    4. `b[x'][y'] = step; x = x'; y = y';`
- 打印b，输出跳马线路.

该算法不能保证对任意的起点都有解存在(算法收敛)，对如8×8的棋盘如果起点为$(5, 4)$将搜索失败. 详见文件warndorff.c.

本章小节

1. 路径
   - 路径与基本路径
   - 连通性
   - 回路与基本回路
   - 距离与直径

2. 路径的应用
   - 赋权图与最短路径
   - Euler路径与回路
   - Hamilton路径与回路