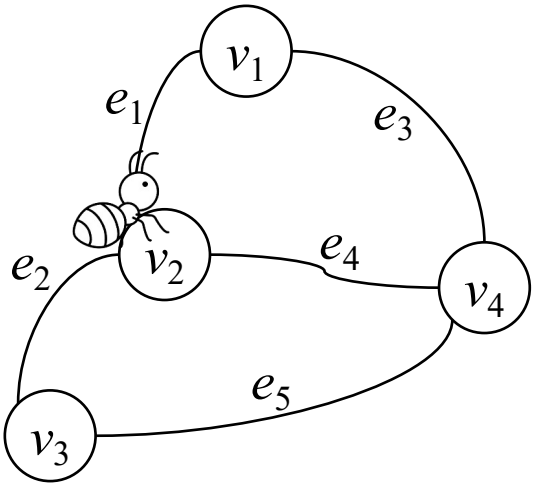
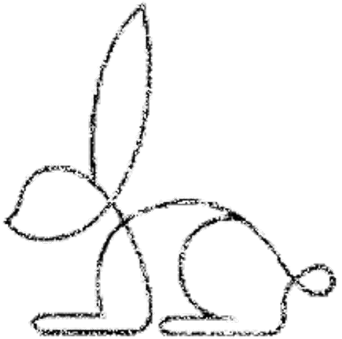
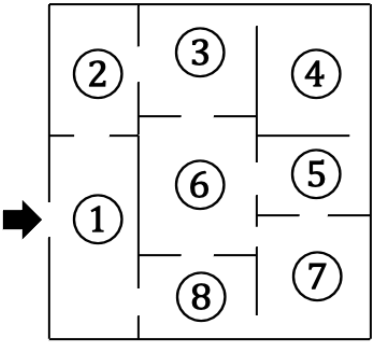
3-3 圈和遍历

程龚

与上次课的遍历有什么区别?

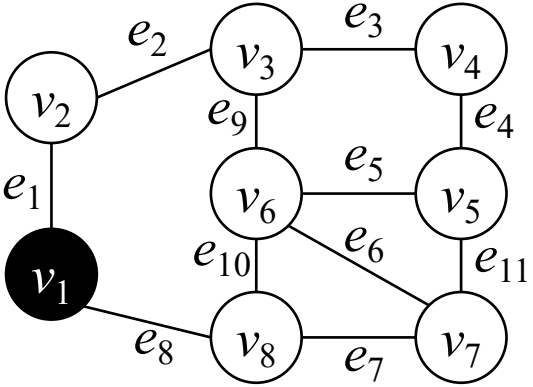






你能解释这些术语吗？

- 闭路线
- 闭迹
- 圈
- 围长、周长

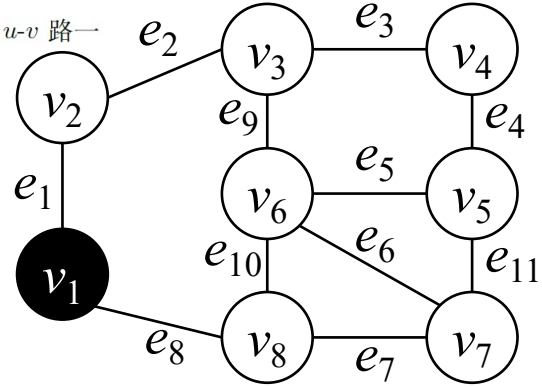


你能解释这些术语吗？

- 闭路线
- 闭迹
- 圈
- 围长、周长

思考题 3.1 若图中存在闭路线，则一定存在闭迹吗？若图中存在闭迹，则一定存在圈吗？

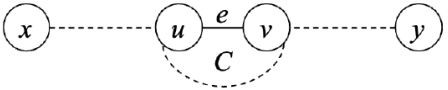
思考题 3.2 含顶点 u 和 v 的圈一定由两条 u-v 路组成吗？两条 u-v 路一定能组成圈吗？



请证明下述定理

定理 3.1 对于图 $G = \langle V, E \rangle$ 和边 $e \in E$，e 是 G 的割边当且仅当 e 不在任何圈中。

- 必要性（反证法）

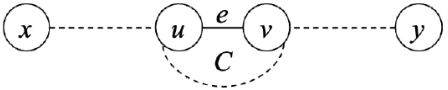


- 充分性（反证法）

它的孪生定理成立吗？

定理 3.1 对于图 $G = \langle V, E \rangle$ 和边 $e \in E$，e 是 G 的割边当且仅当 e 不在任何圈中。

- 必要性（反证法）



- 充分性（反证法）

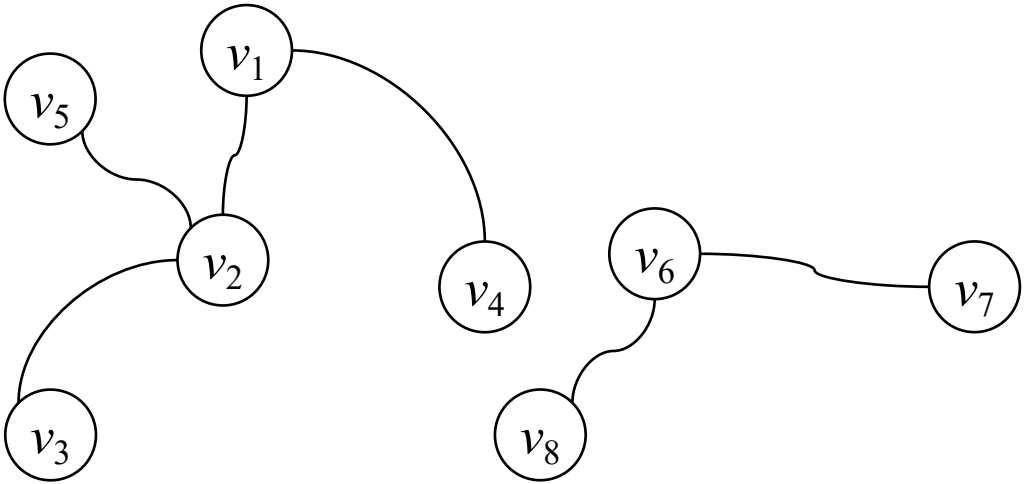
思考题 3.4 对于图 $G = \langle V, E \rangle$ 和顶点 $v \in V$，v 是 G 的割点当且仅当 v 不在任何圈中。这个结论成立吗？

你能解释这些术语吗？

- 树

  思考题 3.7 完全图是树吗？

- 叶顶点
- 森林



树的等价定义

1. G 连通且不含圈。
2. G 中任意两个顶点间有且只有一条路。
3. G 不含圈且 $\epsilon(G) = \nu(G) - 1$。
4. G 连通且 $\epsilon(G) = \nu(G) - 1$。
5. G 极小连通，即：G 连通，但删除任意一条边均不连通。
6. G 极大无圈，即：G 不含圈，但增加任意一条边均形成圈。

树的等价定义

1. G 连通且不含圈。
2. G 中任意两个顶点间有且只有一条路。
3. G 不含圈且 $\epsilon(G) = \nu(G) - 1$。
4. G 连通且 $\epsilon(G) = \nu(G) - 1$。
5. G 极小连通，即：G 连通，但删除任意一条边均不连通。
6. G 极大无圈，即：G 不含圈，但增加任意一条边均形成圈。

思考题 3.9 树中的边有什么特征？你能就此给出树的另一种等价定义吗？

思考题 3.10 非平凡树的叶顶点数量的上界和下界分别是多少？

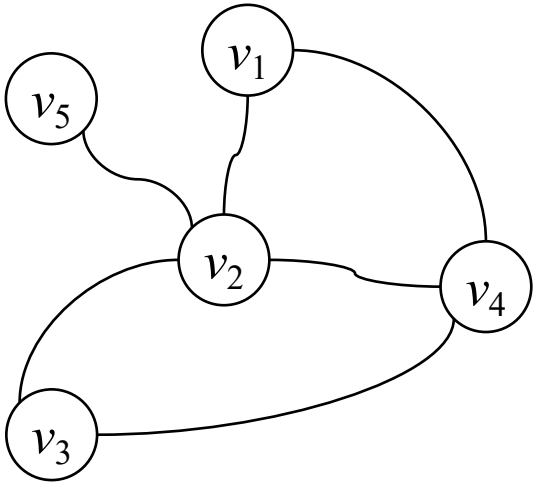
思考题 3.11 对于图 G，若 $\epsilon(G) = \nu(G) - 1$，则 G 一定是树吗？

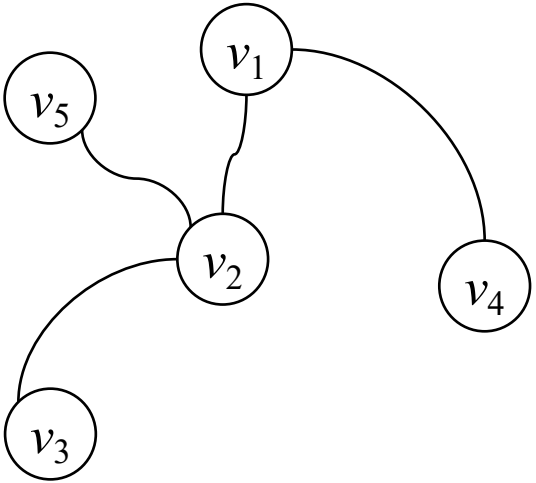
你能解释这些术语吗？

- 生成树

思考题 3.14 连通图一定有生成树吗？生成树唯一吗？生成树唯一的充要条件是什么？

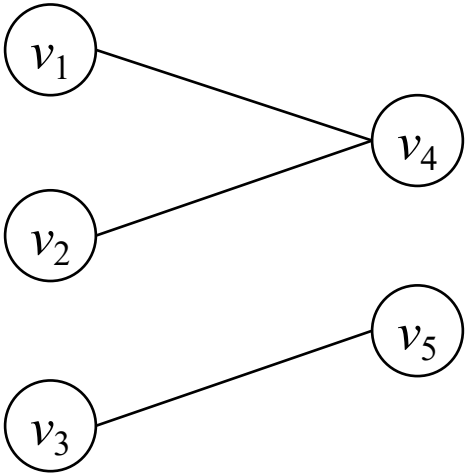
思考题 3.15 DFS 树和 BFS 树是生成树吗？

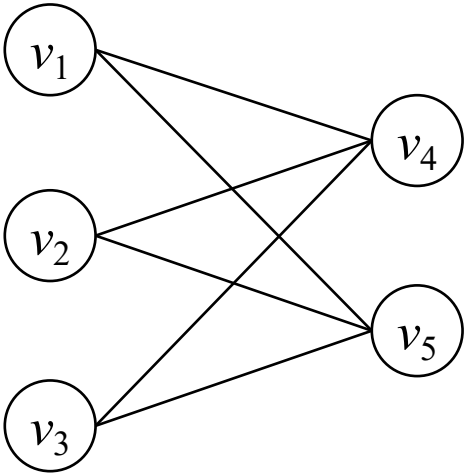




你能解释这些术语吗？

- 奇圈、偶圈
- 二分图
- 完全二分图





你能解释这些术语吗？

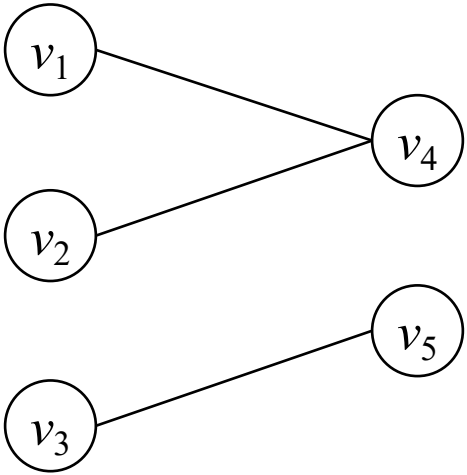
- 奇圈、偶圈
- 二分图
- 完全二分图

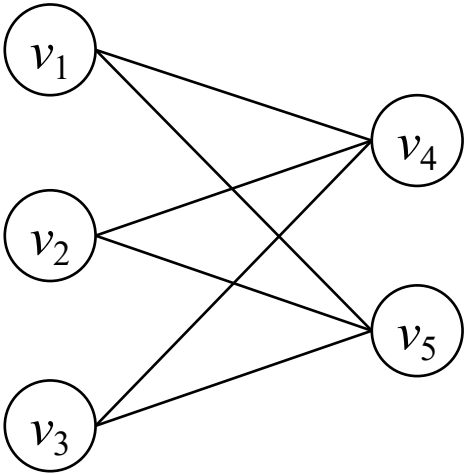
思考题 3.17 二分图的邻接矩阵有什么特征？

思考题 3.18 平凡图是二分图吗？

思考题 3.19 阶为 n 的二分图的边数的上界是多少？

思考题 3.20 对于正则二分图 $G = \langle X \cup Y, E \rangle$，请比较 $|X|$ 和 $|Y|$。

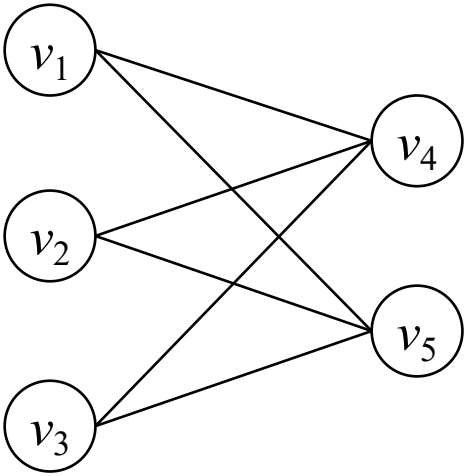


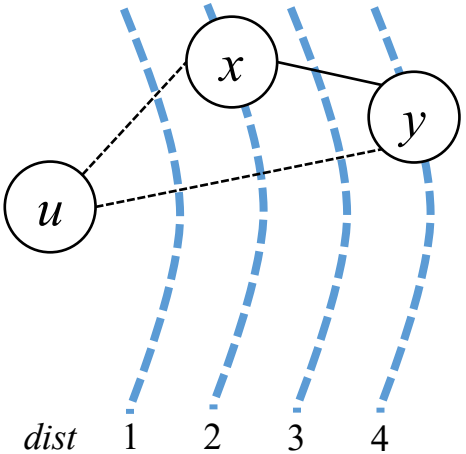


请证明下述充要条件

定理 3.3 非平凡图 G 是二分图当且仅当 G 不含奇圈。

- 必要性
- 充分性（反证法）





请证明下述充要条件

定理 3.3 非平凡图 G 是二分图当且仅当 G 不含奇圈。

思考题 3.23 完全图是二分图吗?

思考题 3.24 树是二分图吗? 森林呢?

思考题 3.25 ($^\diamond$) 对于二分图 $G = \langle V, E \rangle$，若 G 连通，则顶点集 V 的划分方式唯一吗?

思考题 3.26 对于二分图 $G = \langle V, E \rangle$，若顶点集 V 的划分方式唯一，则 G 一定连通吗?

你理解这个算法了吗？

- samePart的含义？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```

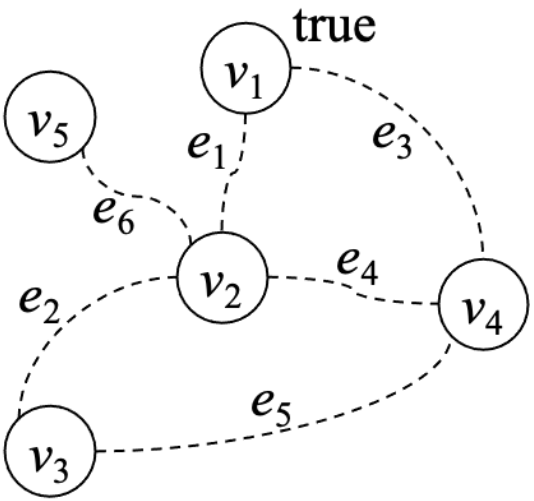
你理解这个算法了吗？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



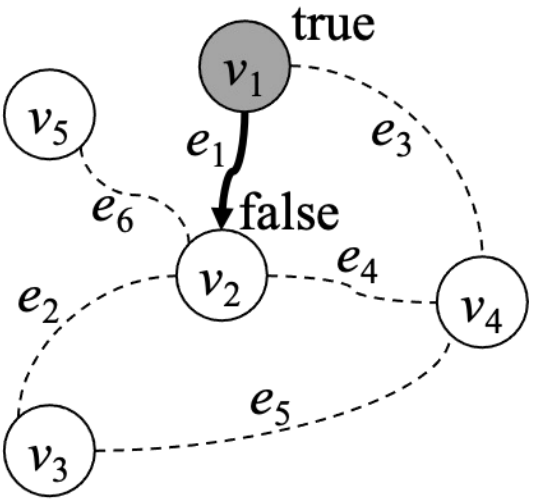
你理解这个算法了吗？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



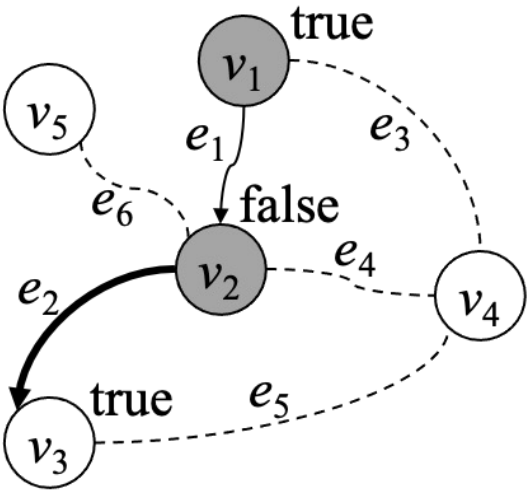
你理解这个算法了吗？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



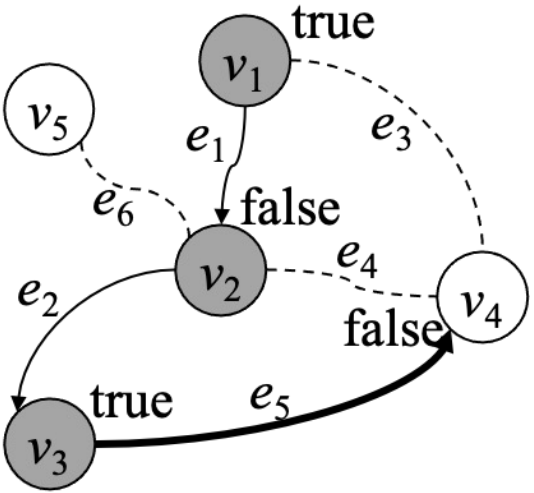
你理解这个算法了吗？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



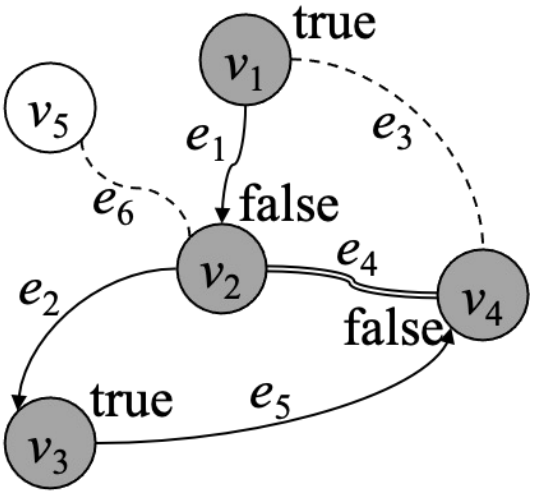
你理解这个算法了吗？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



你理解这个算法了吗？

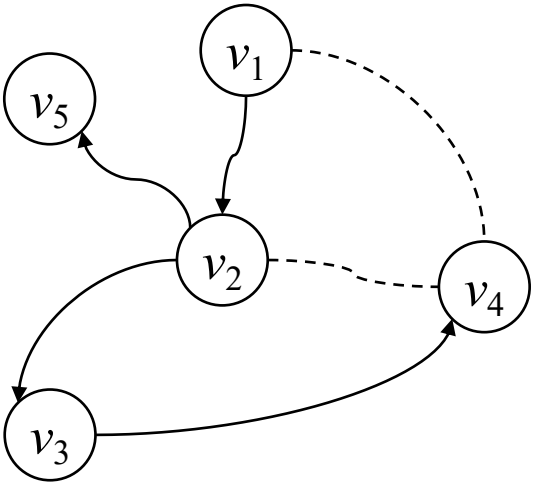
思考题 3.27 在 DFSBpt 算法中，树边和后向边的作用分别是什么？

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



请证明下述定理

定理 3.4 DFSBpt 算法的判定结果正确。

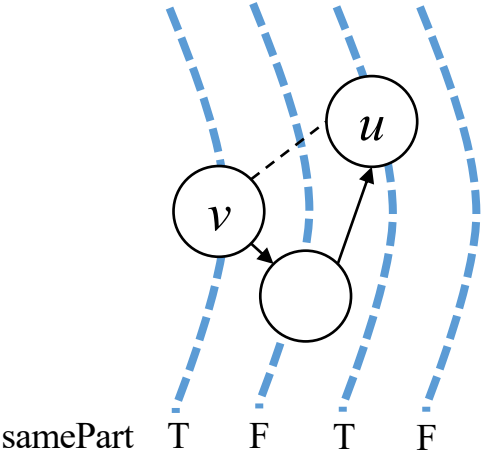
- 若判定为二分图
- 若判定为非二分图

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```



你能解释这个算法的时间复杂度吗？

- 对于阶为n、边数为m的图，DFSBpt算法的时间复杂度为$O(n+m)$

算法 3.1: DFSBpt

输入: 非平凡连通图 $G = \langle V, E \rangle$，顶点 u

初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        v.samePart ← ¬ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart then
        判定（“G 非二分图”）
```

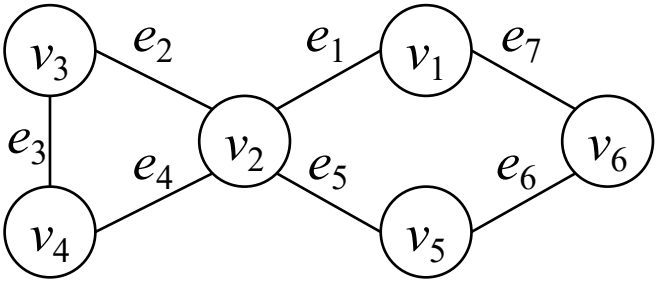
你能解释这些术语吗？

- 欧拉迹
- 欧拉回路
- 欧拉图

思考题 3.29 每个图都有欧拉迹吗？若有，则唯一吗？

思考题 3.30 树是欧拉图吗？

思考题 3.31 对于连通欧拉图 $G = \langle V_G, E_G \rangle$ 和 $H = \langle V_H, E_H \rangle$，任取顶点 $u \in V_G$ 和 $v \in V_H$，向 G 和 H 的不交并 $G + H$ 中增加边 (u, v) 形成的图含欧拉回路吗？含欧拉迹吗？

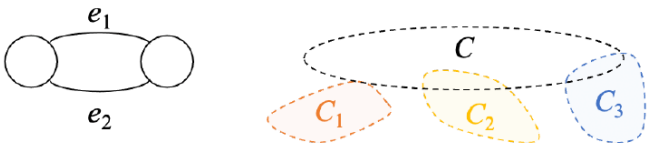


请证明下述定理和推论

定理 3.5 非空连通图 G 含欧拉回路当且仅当 G 没有顶点的度为奇数。

推论 3.1 非空连通图 G 含欧拉迹当且仅当 G 有至多 2 个顶点的度为奇数。

- 必要性
- 充分性（数学归纳法）



(a) (b)

- 必要性
- 充分性

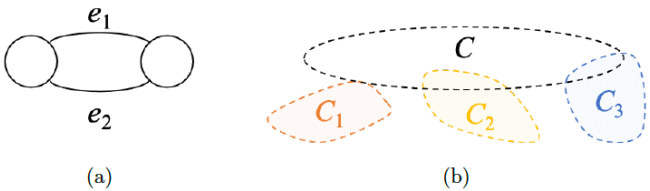
思考题 3.33 含欧拉迹的图一定连通吗？图 G 含欧拉回路和欧拉迹的充要条件分别是什么？

请证明下述定理和推论

定理 3.5 非空连通图 G 含欧拉回路当且仅当 G 没有顶点的度为奇数。

推论 3.1 非空连通图 G 含欧拉迹当且仅当 G 有至多 2 个顶点的度为奇数。

- 必要性
- 充分性（数学归纳法）



思考题 3.36 完全图是欧拉图吗?

思考题 3.37 完全二分图是欧拉图吗?

思考题 3.38 正则图是欧拉图吗?

- 必要性
- 充分性

思考题 3.33 含欧拉迹的图一定连通吗？图 G 含欧拉回路和欧拉迹的充要条件分别是什么?

你理解这个算法了吗？

- 如何选择出发点？
- 哪一步的时间复杂度最高？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

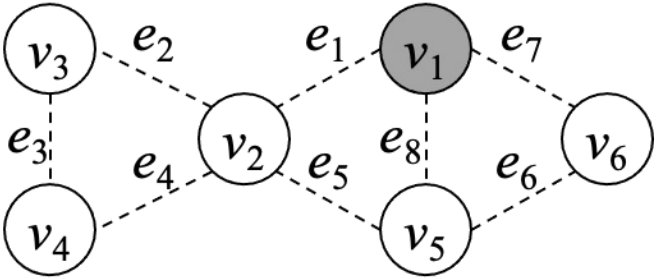
```
if ∃v ∈ V, d(v)是奇数 then
    u ← v;
else
    u ← V 中任意一个非孤立点;
输出 (u);
while u 非孤立点 do
    if u 关联一条非割边 e′ then
        e ← e′;
    else
        e ← u 关联的任意一条边;
    输出 (e);
    u ← e 的另一个端点;
    输出 (u);
    G ← G − e;
```

你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

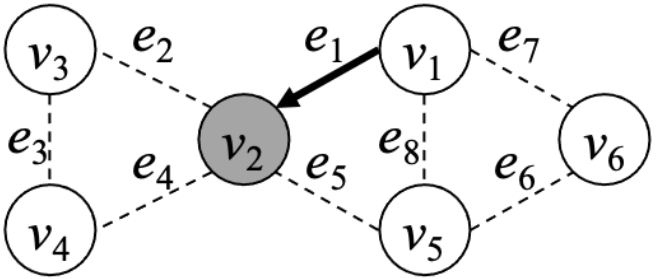


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

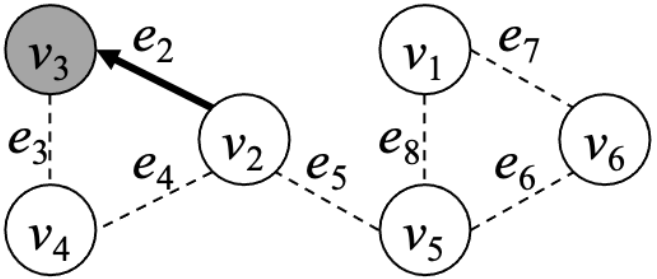


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

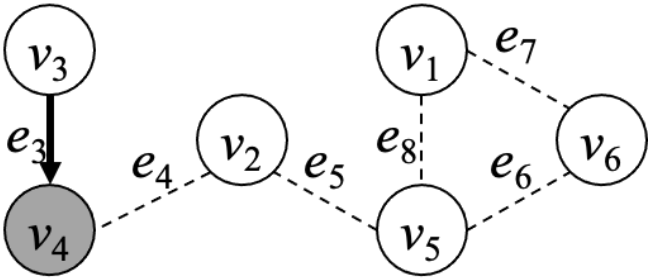


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

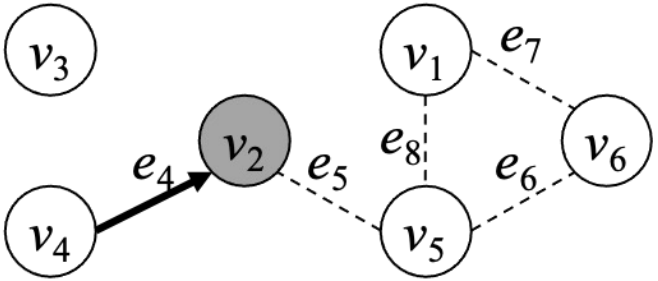


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

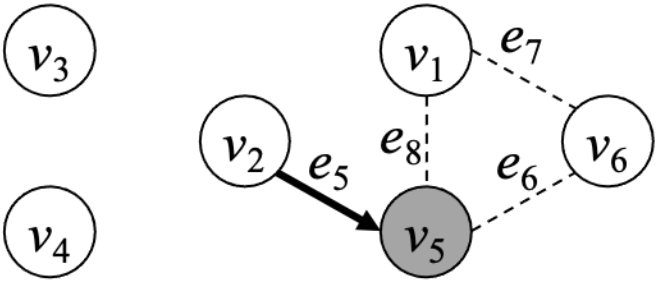


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

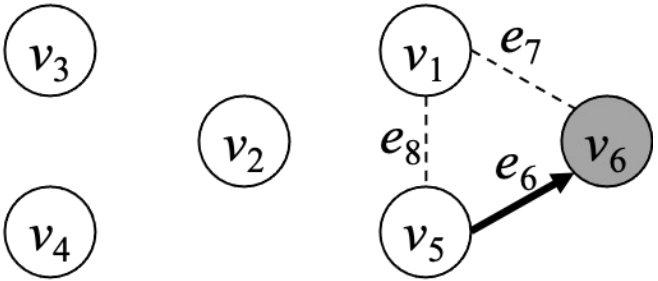


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

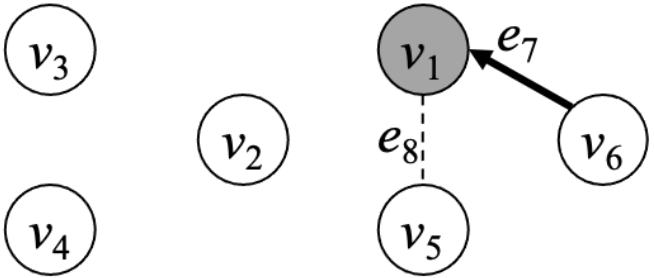


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

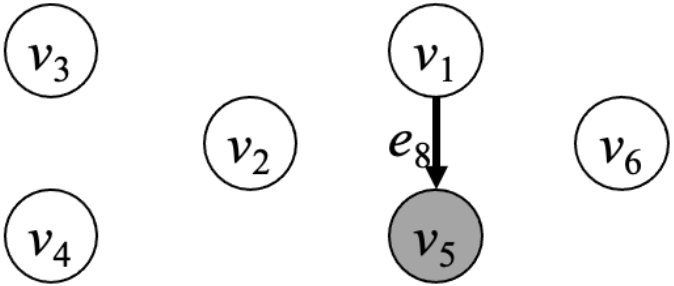


你理解这个算法了吗？

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists v \in V,\ d(v)$是奇数 **then**
  $u \leftarrow v$;
else
  $u \leftarrow V$ 中任意一个非孤立点;
输出 (u);
while u 非孤立点 **do**
  if u 关联一条非割边 e' **then**
    $e \leftarrow e'$;
  else
    $e \leftarrow u$ 关联的任意一条边;
  输出 (e);
  $u \leftarrow e$ 的另一个端点;
  输出 (u);
  $G \leftarrow G - e$;

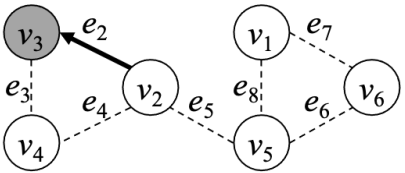


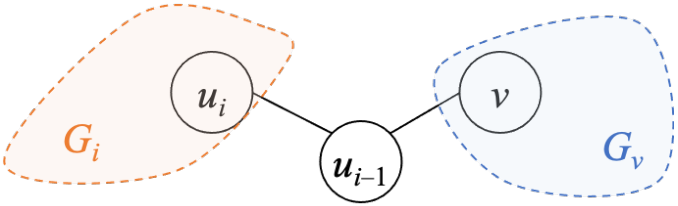
请证明下述定理和推论

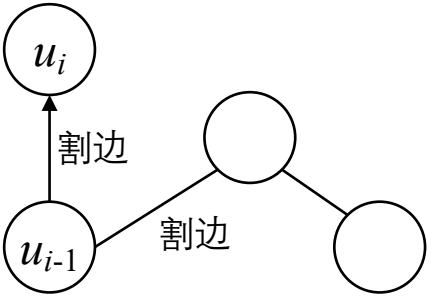
- 算法每轮while循环开始前：
  - 或者图G为空图，则原图的欧拉迹已找到；
  - 或者G含以顶点u为起点的欧拉迹，则该欧拉迹和已输出的序列可拼接得到原图的欧拉迹。

定理 3.6 弗勒里算法每轮 while 循环条件判定前，图 G 或者没有顶点的度为奇数，或者恰有 2 个顶点（包括顶点 u）的度为奇数。

推论 3.2 弗勒里算法每轮 while 循环条件判定前，图 G 或者为空图，或者边集的边导出子图连通且含顶点 u。







你能解释这个算法的时间复杂度吗？

- 对于阶为n、边数为m的图，弗勒里算法的时间复杂度为$O(n+m(n+m))$

算法 3.2: 弗勒里算法

输入: 含欧拉迹的图 $G=\langle V,E\rangle$

```
if ∃v ∈ V, d(v)是奇数 then
    u ← v;
else
    u ← V 中任意一个非孤立点;
输出 (u);
while u 非孤立点 do
    if u 关联一条非割边 e′ then
        e ← e′;
    else
        e ← u 关联的任意一条边;
    输出 (e);
    u ← e 的另一个端点;
    输出 (u);
    G ← G − e;
```

你理解这个算法了吗？

- 如何选择第一条迹？
- 要找的迹一定存在吗？如何找出这样的迹？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**
  $T \leftarrow G$ 中任意一条 u-v 迹;
else
  $T \leftarrow G$ 中任意一条闭迹;
$G \leftarrow G - T$经过的边的集合;
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**
  $T' \leftarrow G$ 中任意一条 w-w 闭迹;
  $T \leftarrow T$ 和 T' 拼接;
  $G \leftarrow G - T'$经过的边的集合;
输出 (T)

你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

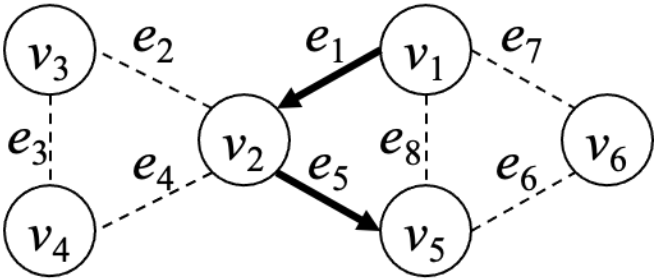
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

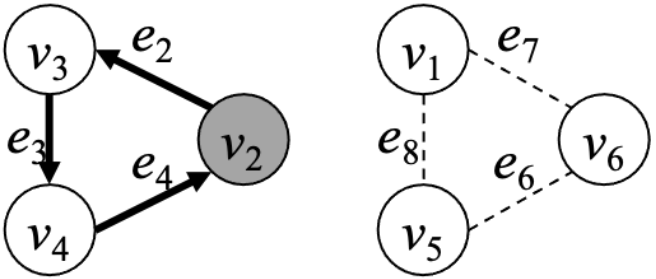
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

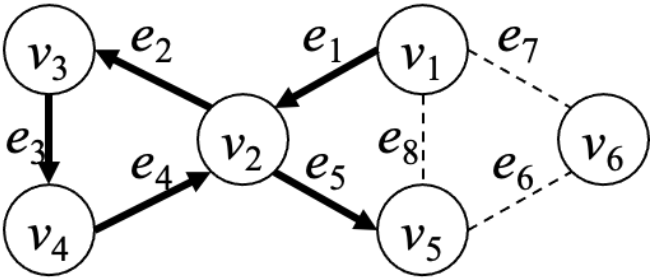
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

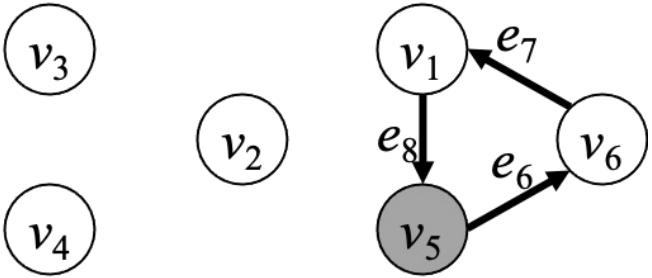
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

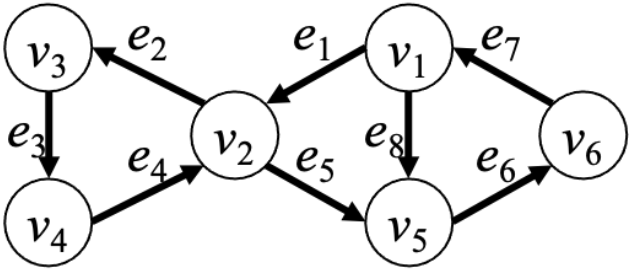
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



你理解这个算法了吗？

算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V,\ d(u)$是奇数 且 $d(v)$是奇数 **then**

  $T \leftarrow G$ 中任意一条 u-v 迹;

else

  $T \leftarrow G$ 中任意一条闭迹;

$G \leftarrow G - T$经过的边的集合;

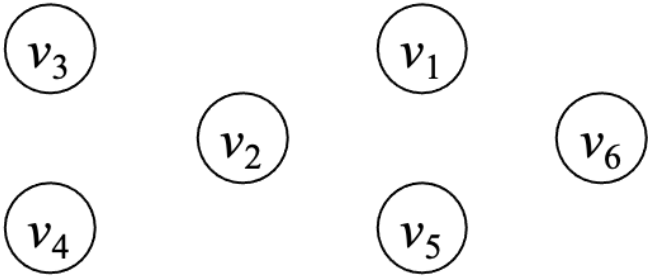
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**

  $T' \leftarrow G$ 中任意一条 w-w 闭迹;

  $T \leftarrow T$ 和 T' 拼接;

  $G \leftarrow G - T'$经过的边的集合;

输出 (T)



请证明下述定理

- 算法每轮while循环开始前：
  - 或者图G为空图，则原图的欧拉迹已找到；
  - 或者G的每个非平凡连通分支都含（该分支自身的）欧拉回路，且都和迹T有公共顶点，则这些欧拉回路和T可拼接得到原图的欧拉迹。

定理 3.7 希尔霍尔策算法每轮 while 循环条件判定前，图 G 没有顶点的度为奇数。

定理 3.8 希尔霍尔策算法每轮 while 循环条件判定前，图 G 或者为空图，或者边集的边导出子图的每个连通分支都和迹 T 有公共顶点。

你能解释这个算法的时间复杂度吗？

- 对于阶为n、边数为m的图，希尔霍尔策算法的时间复杂度为$O(n+m)$

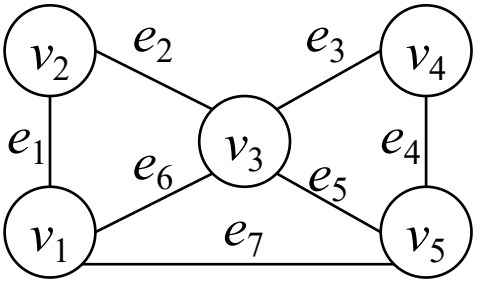
算法 3.3: 希尔霍尔策算法

输入: 含欧拉迹的图 $G = \langle V, E \rangle$

if $\exists u, v \in V$, $d(u)$是奇数 且 $d(v)$是奇数 **then**
  $T \leftarrow G$ 中任意一条 u-v 迹;
else
  $T \leftarrow G$ 中任意一条闭迹;
$G \leftarrow G - T$经过的边的集合;
while $\exists w \in V$, w 非孤立点且 T 经过 w **do**
  $T' \leftarrow G$ 中任意一条 w-w 闭迹;
  $T \leftarrow T$ 和 T' 拼接;
  $G \leftarrow G - T'$经过的边的集合;
输出 (T)

你能解释这些术语吗？

- 哈密尔顿路
- 哈密尔顿圈
- 哈密尔顿图



思考题 3.42 每个图都有哈密尔顿路和哈密尔顿圈吗？若有，则唯一吗？

思考题 3.43 树是哈密尔顿图吗？

思考题 3.45 完全图是哈密尔顿图吗？

思考题 3.46 完全二分图是哈密尔顿图吗？

思考题 3.47 对于哈密尔顿图 $G = \langle V_G, E_G \rangle$ 和 $H = \langle V_H, E_H \rangle$，任取顶点 $u \in V_G$ 和 $v \in V_H$，向 G 和 H 的不交并 $G + H$ 中增加边 (u, v) 形成的图是哈密尔顿图吗？含哈密尔顿路吗？

思考题 3.48 哈密尔顿图一定连通吗？有割点和割边吗？

思考题 3.49 欧拉图是哈密尔顿图吗？哈密尔顿图是欧拉图吗？

哈密尔顿路/圈的存在性的判定问题如何互相归约?

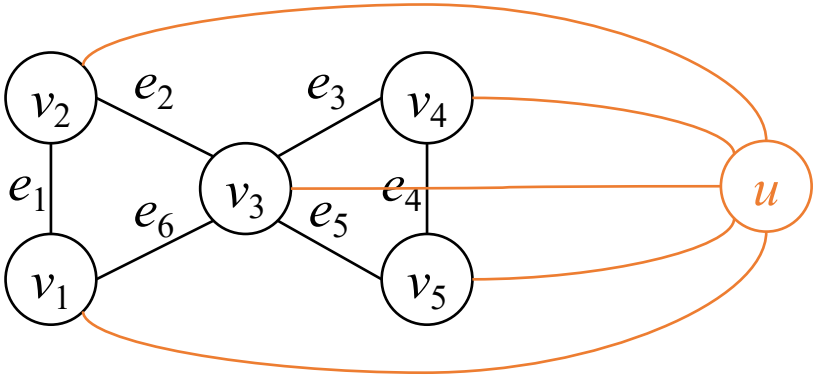
思考题 3.53 对于图 G，如何构造图 H，使 G 含哈密尔顿路当且仅当 H 含哈密尔顿圈?

思考题 3.54 ($^{\diamond}$) 对于图 G，如何构造图 H，使 G 含哈密尔顿圈当且仅当 H 含哈密尔顿路?

哈密尔顿路/圈的存在性的判定问题如何互相归约?

思考题 3.53 对于图 G，如何构造图 H，使 G 含哈密尔顿路当且仅当 H 含哈密尔顿圈?

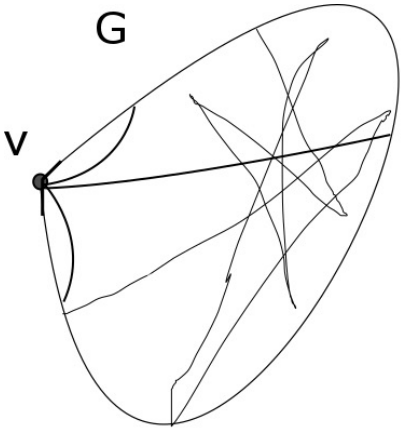
思考题 3.54 ($^\diamond$) 对于图 G，如何构造图 H，使 G 含哈密尔顿圈当且仅当 H 含哈密尔顿路?

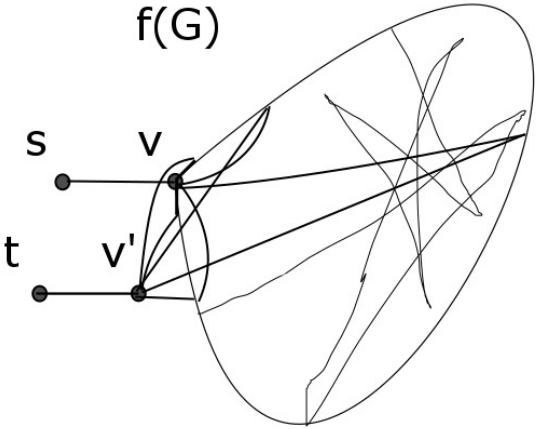


哈密尔顿路/圈的存在性的判定问题如何互相归约？

思考题 3.53 对于图 G，如何构造图 H，使 G 含哈密尔顿路当且仅当 H 含哈密尔顿圈？

思考题 3.54 ($^{\diamond}$) 对于图 G，如何构造图 H，使 G 含哈密尔顿圈当且仅当 H 含哈密尔顿路？





https://math.stackexchange.com/questions/7130/reduction-from-hamiltonian-cycle-to-hamiltonian-path/1290804#1290804

OT

- 请调研教材中未介绍过的至少2种哈密尔顿路或哈密尔顿圈的存在性的必要条件或充分条件，讨论条件的适用场景，并阐述证明过程。