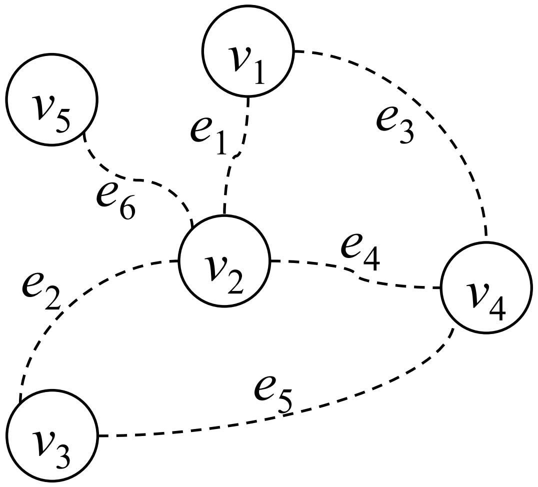
第3章 圈和遍历

程龚
南京大学 计算机学院
gcheng@nju.edu.cn
http://ws.nju.edu.cn/~gcheng

本章内容

- 第3.1节 圈和树
- 第3.2节 二分图
  - 第3.2.1节 理论
  - **第3.2.2节 算法**
- 第3.3节 欧拉图
- 第3.4节 哈密尔顿图

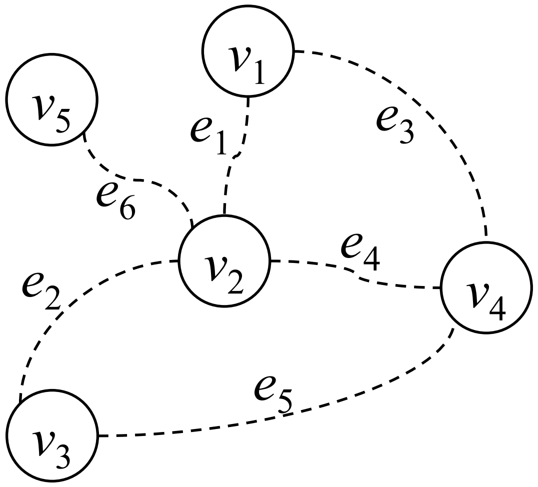
如何判定一个图是否为二分图？



如何判定一个图是否为二分图？

思考题3.25 对于二分图$G = <V, E>$，若G连通，则顶点集V的划分方式唯一吗？

- 唯一：任取一个顶点u，只能按与u间的距离的奇偶性二分



DFSBpt算法

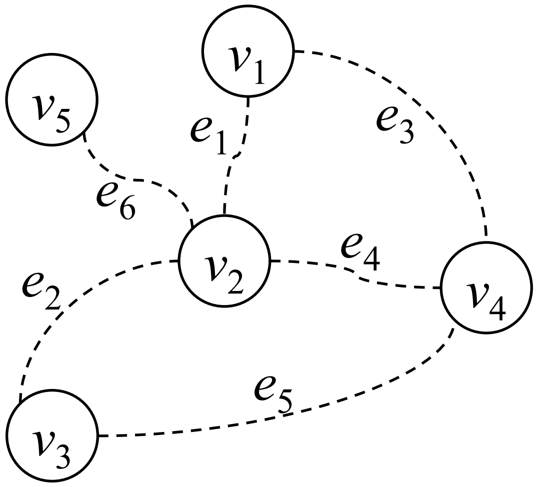
- 基本思路：从非平凡连通图中任意一个指定顶点出发，按DFS的方式有序地遍历所有顶点并尝试将其划分为两个子集X和Y，根据每条边的两个端点是否分属于X和Y，判定该图是否为二分图。

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”）
```



DFSBpt算法

- 基于DFS算法

算法 3.1: DFSBpt

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

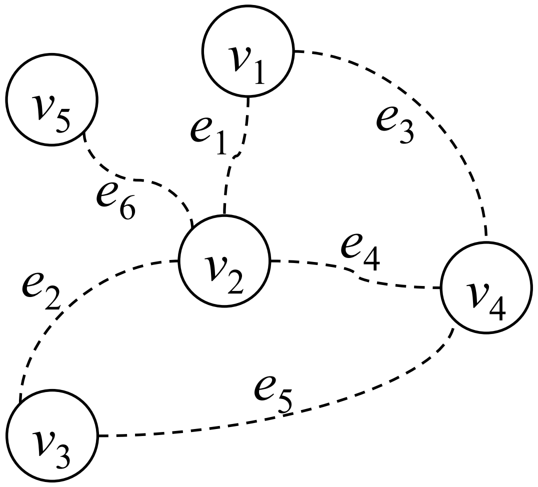
初值: 顶点集 V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

foreach $(u, v) \in E$ **do**

  if v.visited $=$ false **then**

    `DFSBpt(G, v);`



DFSBpt算法

- 扩展：记录每个顶点属于顶点子集X还是Y
  - 不失一般性，假设出发点属于X
  - 每个顶点的samePart属性：布尔型变量，表示该顶点是否和出发点同属于X

算法 3.1: DFSBpt

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

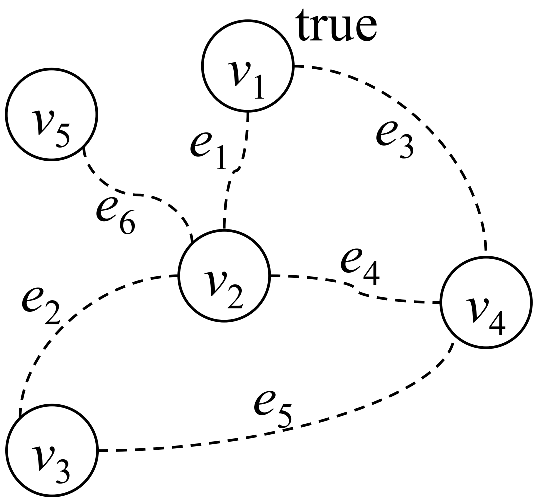
初值: 顶点集 V 中所有顶点的 visited 初值为 false，出发点的 samePart 初值为 true、其它顶点的 samePart 初值未定义

u.visited $\leftarrow$ true;

foreach $(u, v) \in E$ **do**

  if v.visited $=$ false **then**

    DFSBpt(G, v);



DFSBpt算法

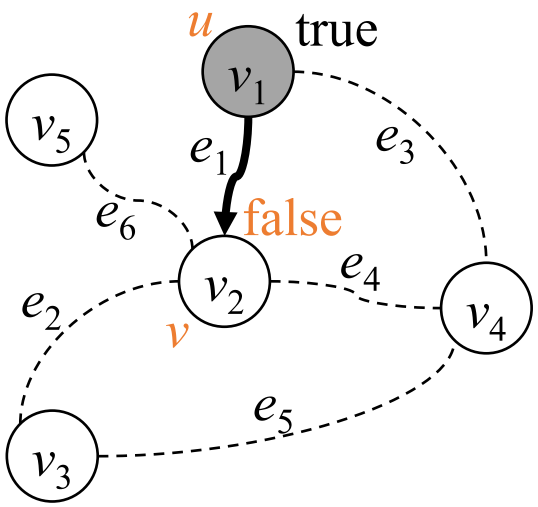
- 扩展：记录每个顶点属于顶点子集X还是Y
  - 不失一般性，假设出发点属于X
  - 每个顶点的samePart属性：布尔型变量，表示该顶点是否和出发点同属于X
  - 对于顶点u的每个邻点v：
    - 若v未被访问过，则在递归调用算法访问v之前，将u的samePart属性值的非作为v的samePart属性值

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 $\leftarrow$ true;
foreach $(u, v) \in E$ **do**
  if v.visited = false **then**
    v.samePart $\leftarrow \neg$ u.samePart;
    DFSBpt(G, v);



DFSBpt算法

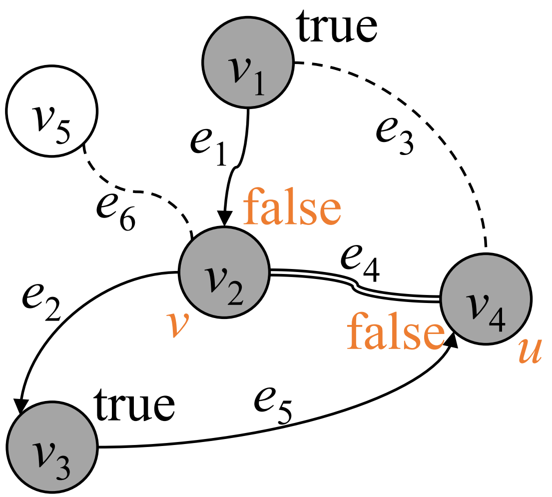
- 扩展：记录每个顶点属于顶点子集X还是Y
  - 不失一般性，假设出发点属于X
  - 每个顶点的samePart属性：布尔型变量，表示该顶点是否和出发点同属于X
  - 对于顶点u的每个邻点v：
    - 若v未被访问过，则在递归调用算法访问v之前，将u的samePart属性值的非作为v的samePart属性值
    - 若v已被访问过，即v已被分到X或Y中，则比较u和v的samePart属性值，若两者相等，则判定G非二分图

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 $\leftarrow$ true;
foreach $(u, v) \in E$ **do**
    if v.visited = false **then**
        v.samePart $\leftarrow \neg$ u.samePart;
        DFSBpt(G, v);
    else if u.samePart = v.samePart **then**
        判定（"G 非二分图"）



DFSBpt算法

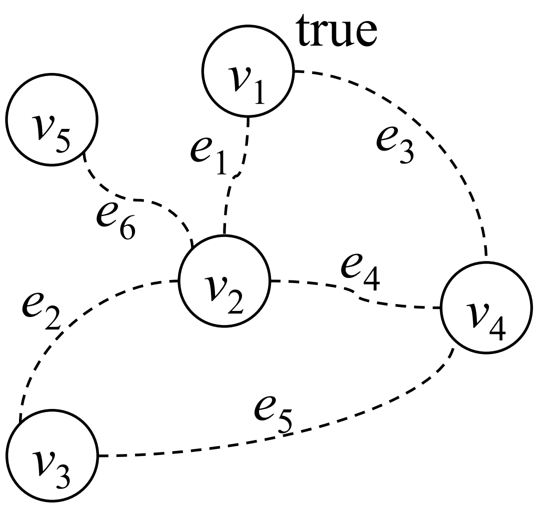
- 例如：从v_1出发
  - v_1.samePart ← true

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 $(u, v) \in E$ **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"）



DFSBpt算法

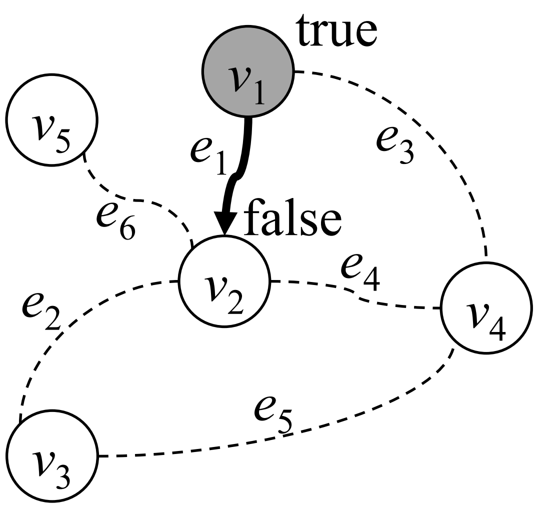
- 调用DFSBpt(G, v_1)
  - v_1.visited ← true
  - 判断v_1的邻点v_2.visited为false，v_2.samePart ← false，递归调用DFSBpt(G, v_2)

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 $(u, v) \in E$ **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”）



DFSBpt算法

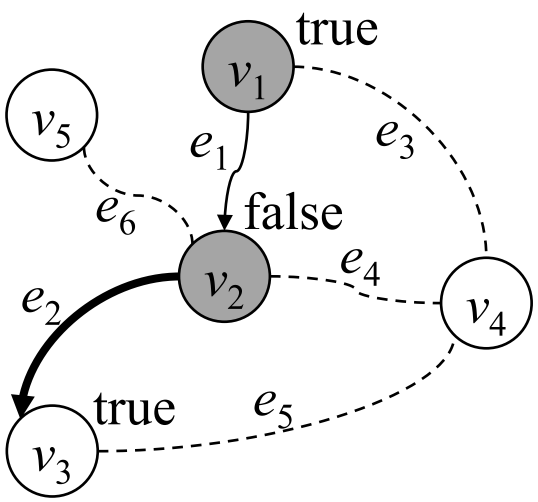
- 调用DFSBpt(G, v_2)
  - v_2.visited ← true
  - 判断v_2的邻点v_3.visited为false， v_3.samePart ← true， 递归调用DFSBpt(G, v_3)

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”）
```



DFSBpt算法

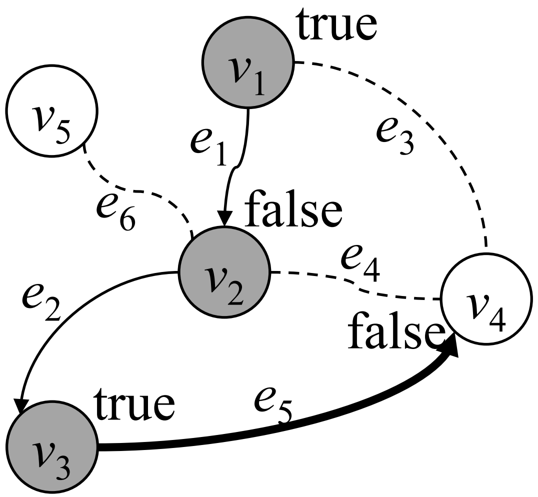
- 调用DFSBpt(G, v_3)
  - v_3.visited ← true
  - 判断v_3的邻点v_4.visited为false， v_4.samePart ← false， 递归调用DFSBpt(G, v_4)

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 $(u, v) \in E$ **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”）



DFSBpt算法

- 调用DFSBpt(G, v_4)
  - v_4.visited ← true
  - 判断v_4的邻点v_2.visited为true，判断v_4和v_2的samePart属性值相等，判定G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 $(u, v) \in E$ **do**

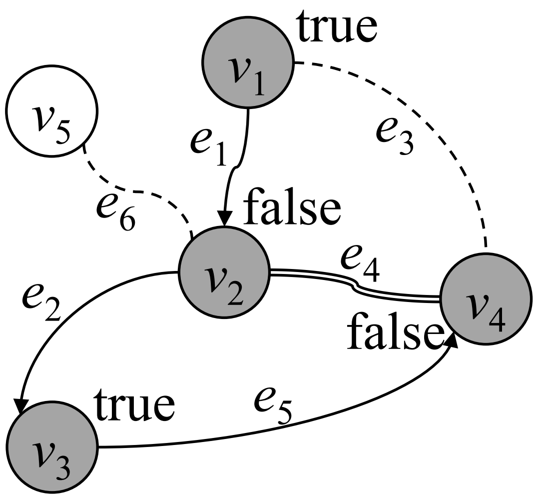
  if v.visited = false **then**

    v.samePart ← ¬ u.samePart;

    DFSBpt(G, v);

  else if u.samePart = v.samePart **then**

    判定（“G 非二分图”）



思考题3.27

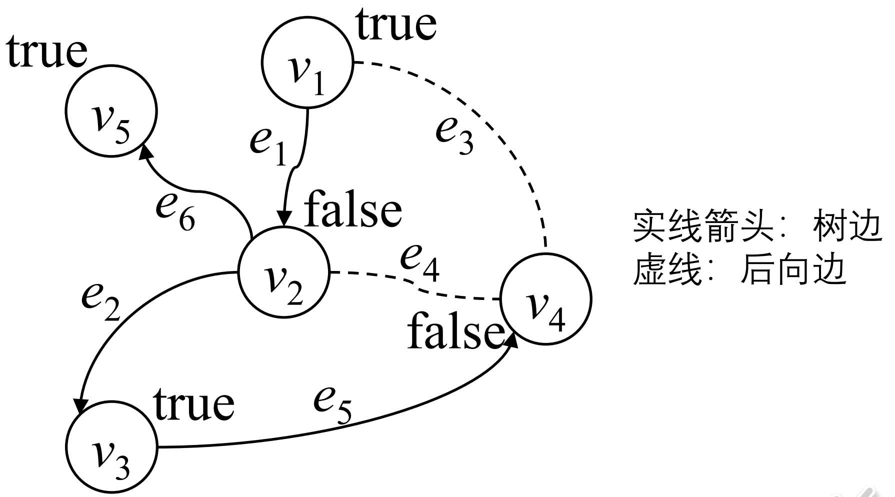
- 在DFSBpt算法中，树边和后向边的作用分别是什么？

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 $(u, v) \in E$ **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”）



思考题3.27

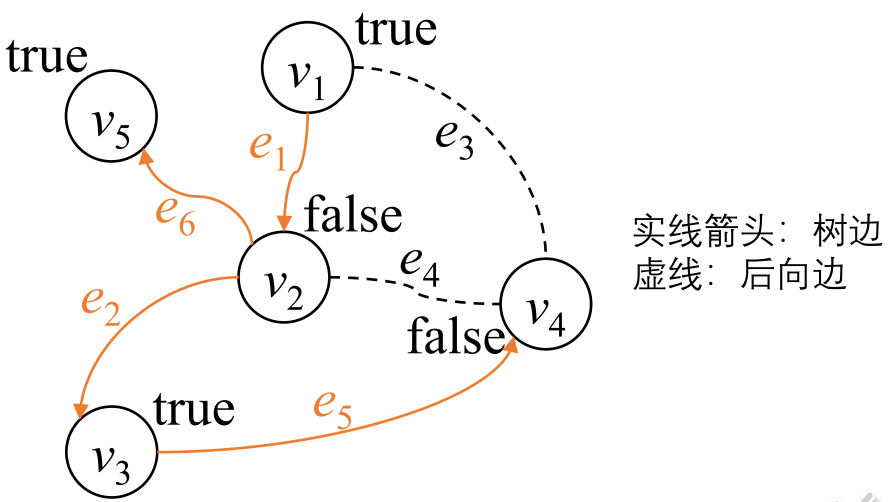
- 在DFSBpt算法中，树边和后向边的作用分别是什么？
  - 树边：将顶点集V划分为两个子集X和Y

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 $(u, v) \in E$ **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"）



思考题3.27

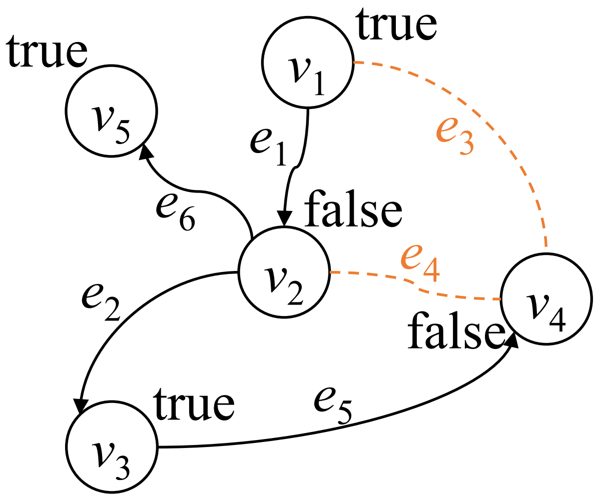
- 在DFSBpt算法中，树边和后向边的作用分别是什么？
  - 树边：将顶点集V划分为两个子集X和Y
  - 后向边：判定圈的长度的奇偶性

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”）
```



实线箭头：树边
虚线：后向边

定理3.4

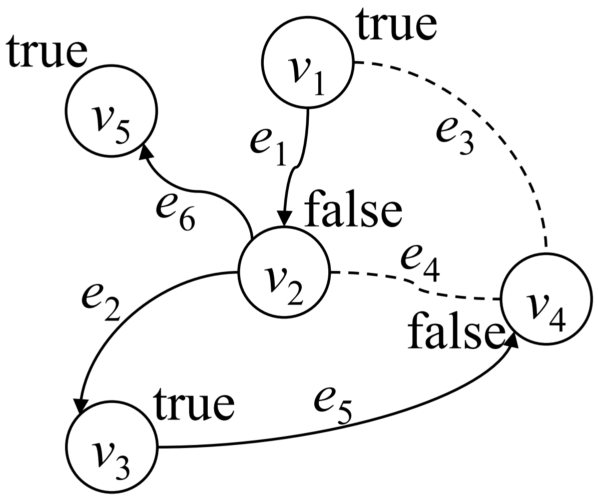
- DFSBpt算法的判定结果正确。

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 $\leftarrow$ true;
foreach $(u, v) \in E$ **do**
    if v.visited = false **then**
        v.samePart $\leftarrow \neg$ u.samePart;
        `DFSBpt(`G, v`);`
    else if u.samePart = v.samePart **then**
        判定（“G 非二分图”）



实线箭头：树边
虚线：后向边

定理3.4

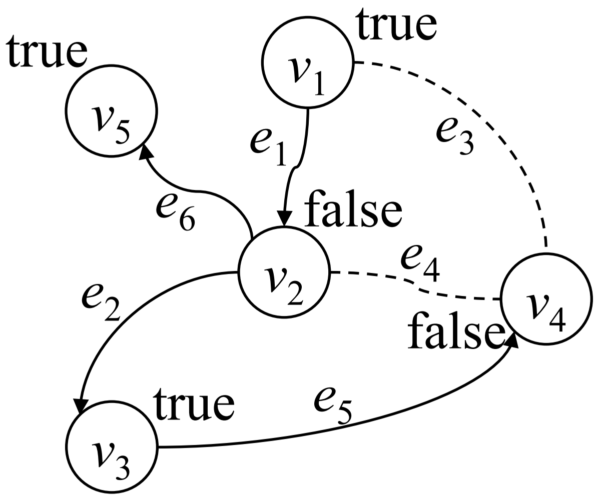
- DFSBpt算法的判定结果正确。
  - 若DFSBpt算法运行结束时判定图G为二分图：
  - 若DFSBpt算法在比较顶点u和v的samePart属性值时判定图G非二分图：

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”）
```



实线箭头：树边
虚线：后向边

定理3.4

对于图$G = <V, E>$，若顶点集V可划分为两个子集X和Y，使每条边$e \in E$的两个端点分属于X和Y，则G称作二分图

- “划分”的含义：$X,Y \neq \emptyset$、$X \cup Y = V$、$X \cap Y = \emptyset$

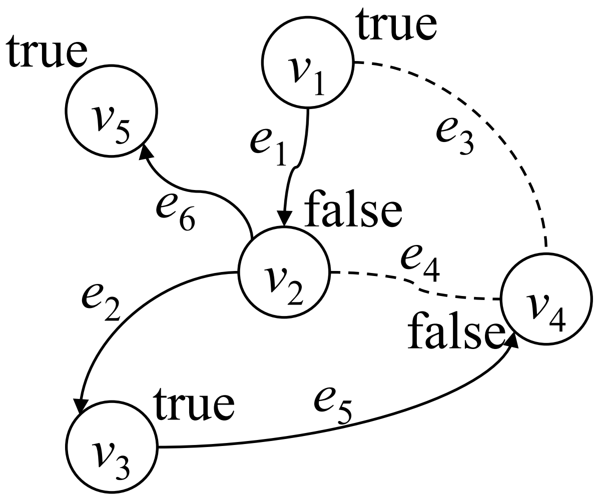
- DFSBpt算法的判定结果正确。
  - 若DFSBpt算法运行结束时判定图G为二分图：
    - 顶点集V已被划分为两个子集：samePart属性值为true的顶点子集X和samePart属性值为false的顶点子集Y。
    - X含出发点，$X \neq \emptyset$；由于G是非平凡连通图，出发点的第一个被访问的邻点的samePart属性值为false，$Y \neq \emptyset$。
    - 算法运行结束时，已检查每条边的两个端点分属于X和Y。
  - 若DFSBpt算法在比较顶点u和v的samePart属性值时判定图G非二分图：

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”）
```



实线箭头：树边
虚线：后向边

定理3.4

引理2.1 后向边关联一对祖先-后代顶点。

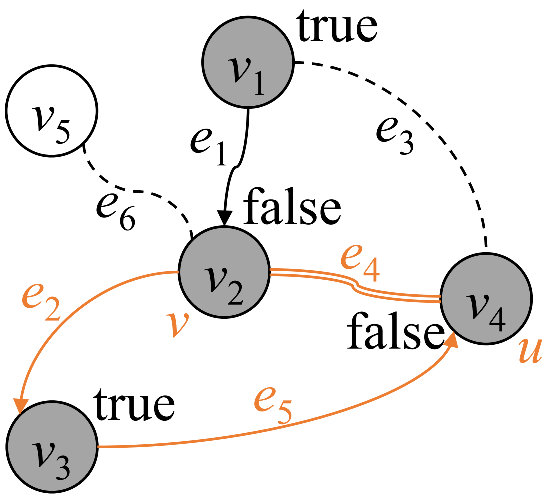
- DFSBpt算法的判定结果正确。
  - 若DFSBpt算法运行结束时判定图G为二分图：
    - 顶点集V已被划分为两个子集：samePart属性值为true的顶点子集X和samePart属性值为false的顶点子集Y。
    - X含出发点，$X \neq \emptyset$；由于G是非平凡连通图，出发点的第一个被访问的邻点的samePart属性值为false，$Y \neq \emptyset$。
    - 算法运行结束时，已检查每条边的两个端点分属于X和Y。
  - 若DFSBpt算法在比较顶点u和v的samePart属性值时判定图G非二分图：
    - 由引理2.1，v是u的祖先顶点。DFS树中的路交替经过samePart属性值为true和false的顶点，
    - 因此，DFS树中v-u路的长度是偶数，该路和后向边(u, v)组成奇圈。

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”）
```



DFSBpt算法

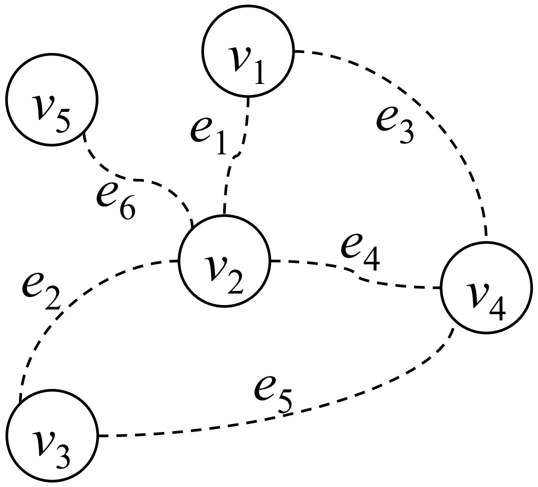
- 时间复杂度：$O(n+m)$

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 $\leftarrow$ true;
foreach $(u,v)\in E$ **do**
  if v.visited $=$ false **then**
    v.samePart $\leftarrow \neg$ u.samePart;
    `DFSBpt(`G, v`);`
  else if u.samePart $=$ v.samePart **then**
    判定（“G 非二分图”）



请认真完成课后练习