连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

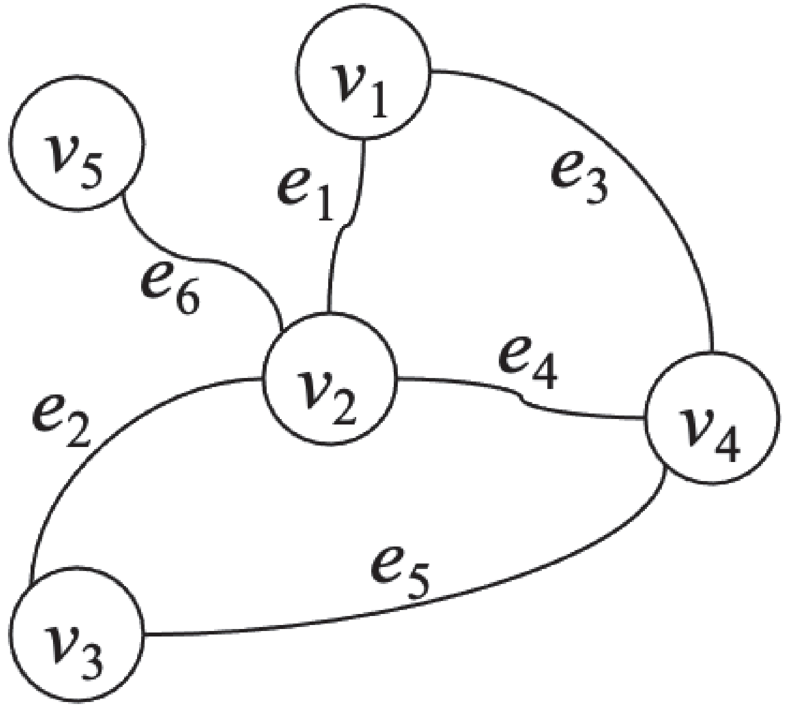
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

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

    DFS(G, v);



连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

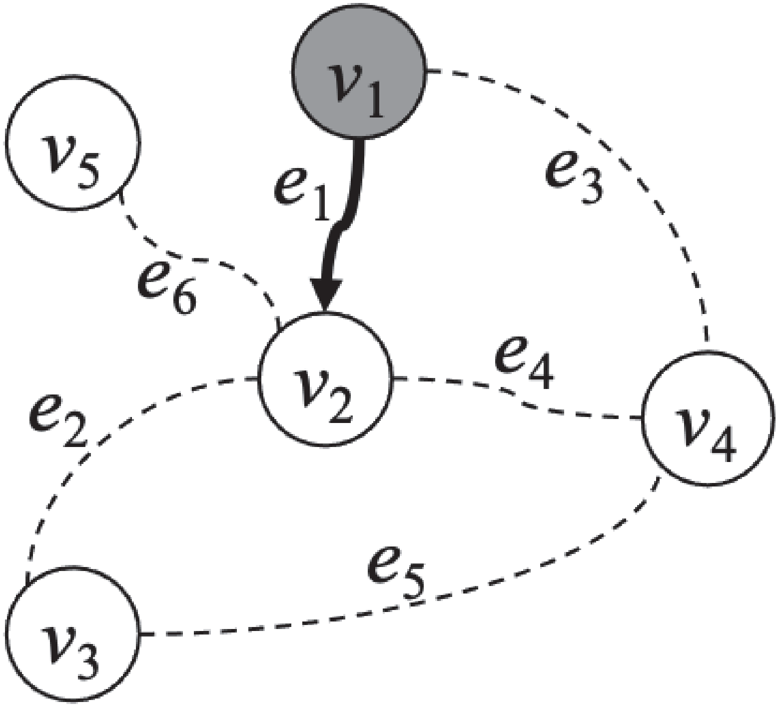
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

    if v.visited = false **then**

        DFS(G, v);



连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

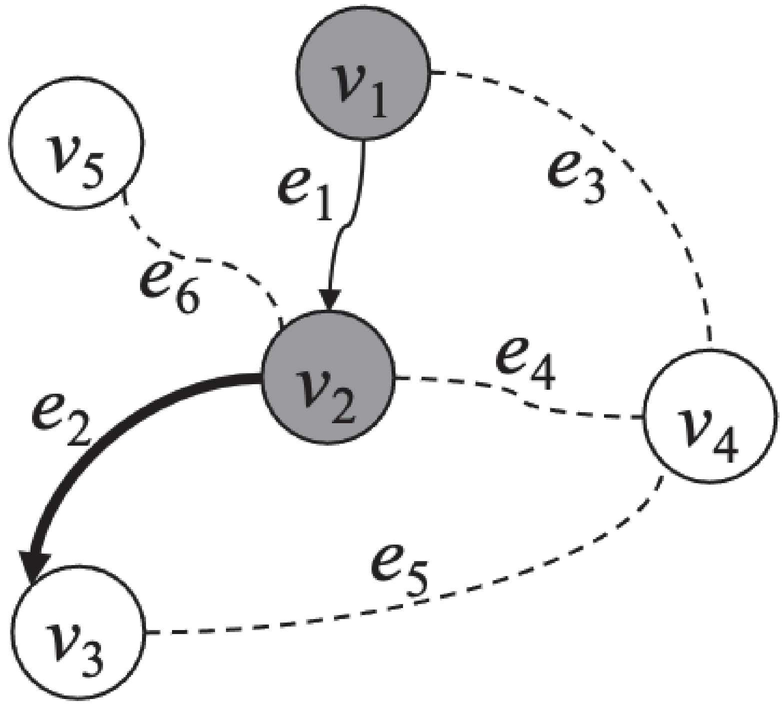
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

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

        `DFS`(G, v);



连通和DFS

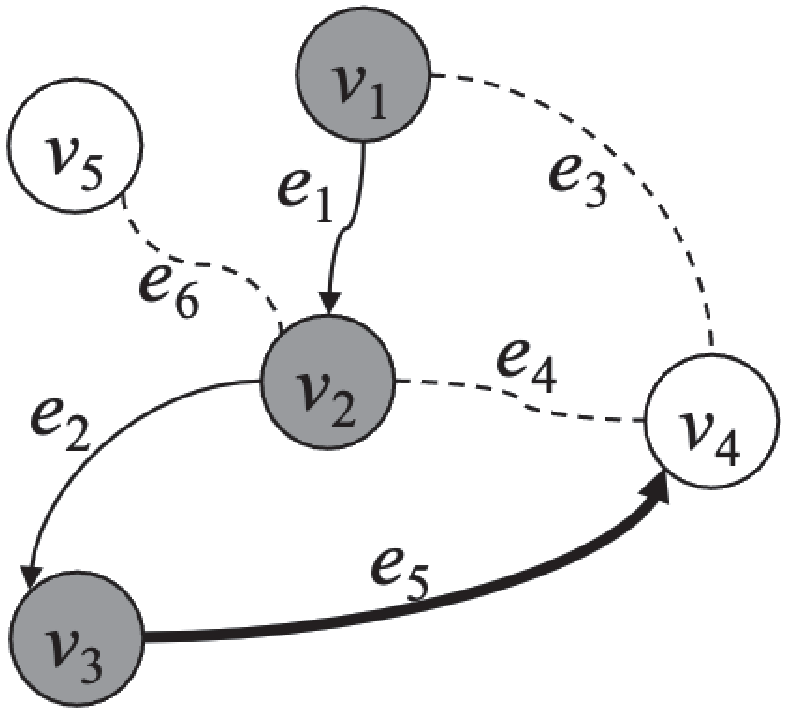
- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

初值: 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**
        DFS(G, v);



连通和DFS

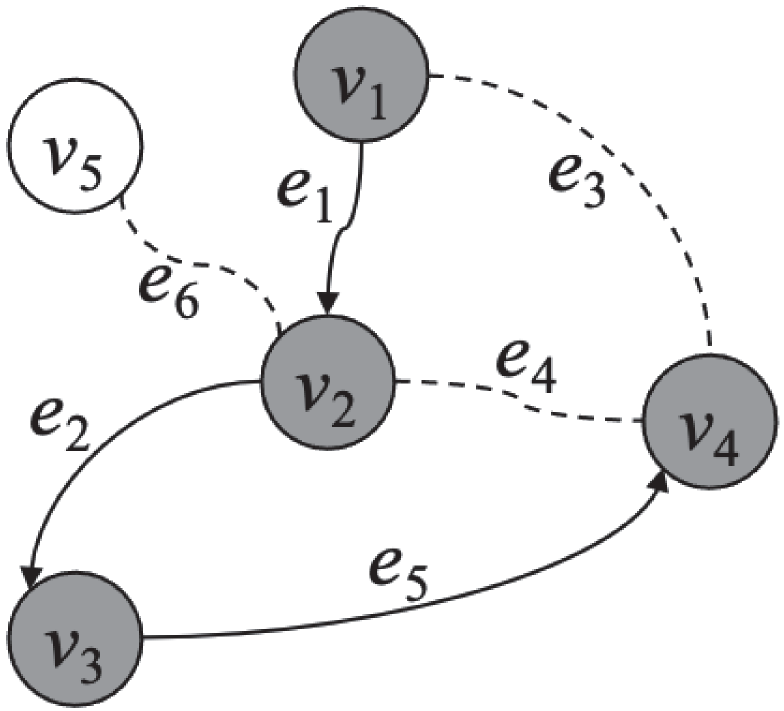
- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

初值: V 中所有顶点的 visited 初值为 false

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        DFS(G, v);
```



连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

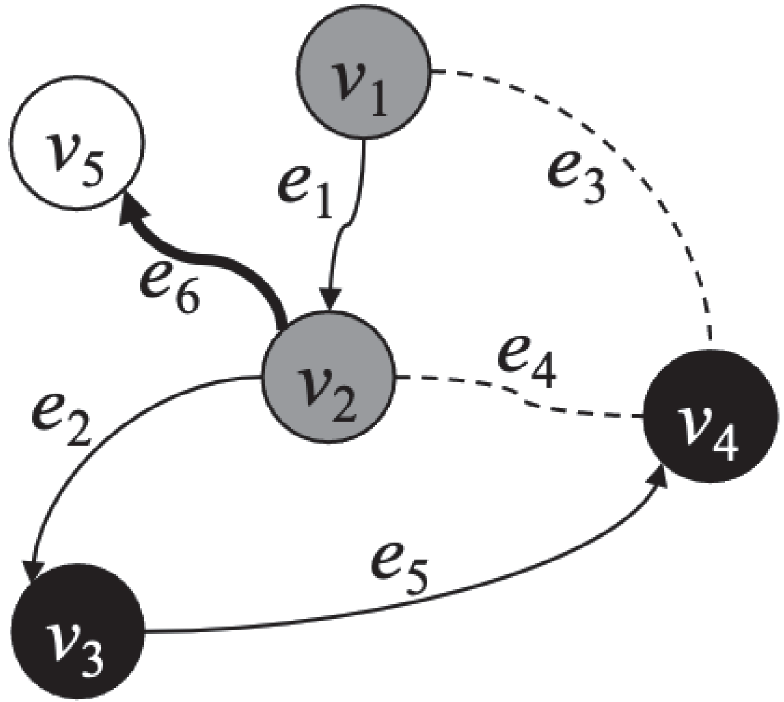
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

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

    DFS(G, v);



连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

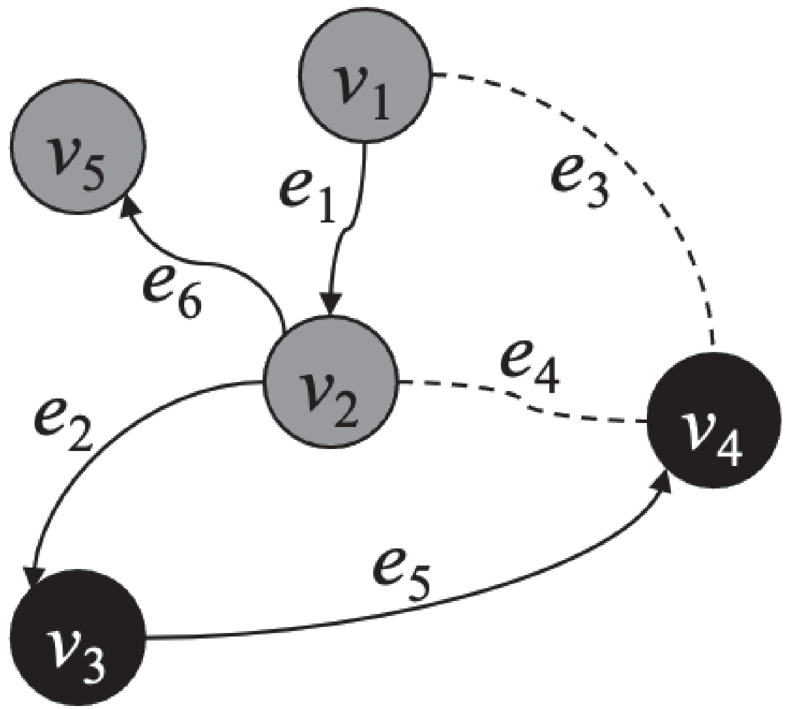
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

  if v.visited = false **then**

    DFS(G, v);



连通和DFS

- 深度优先搜索（DFS）算法
  - 从图中的一个指定顶点出发，有序地遍历和该顶点连通的所有顶点
  - 遍历顶点的顺序是优先向图的“深处”访问，即倾向于远离出发点

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

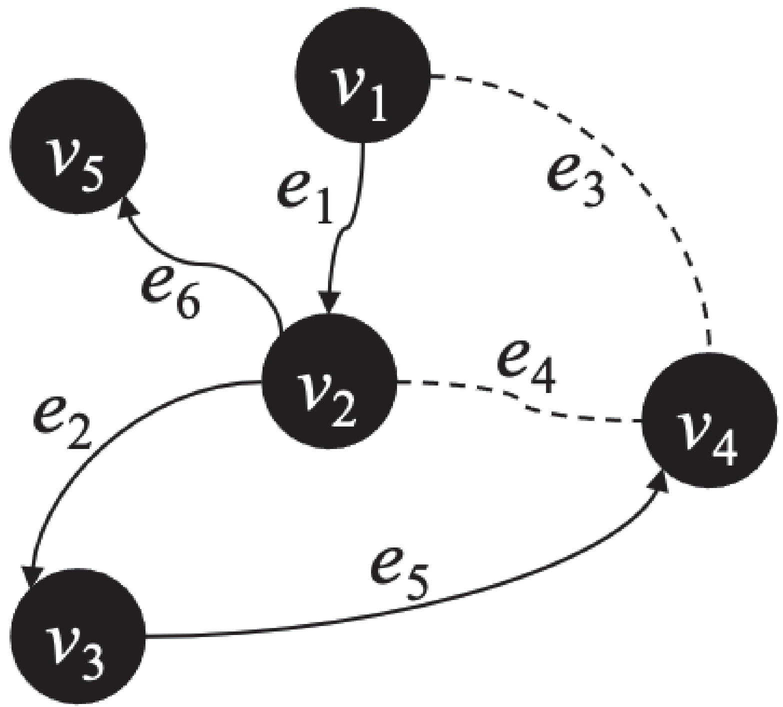
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

  if v.visited = false **then**

    DFS(G, v);



连通和DFS

- 从顶点u出发运行DFS算法，为什么恰能访问与u连通的所有顶点？
  - 为什么连通的都能访问？
  - 为什么访问的都连通？

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

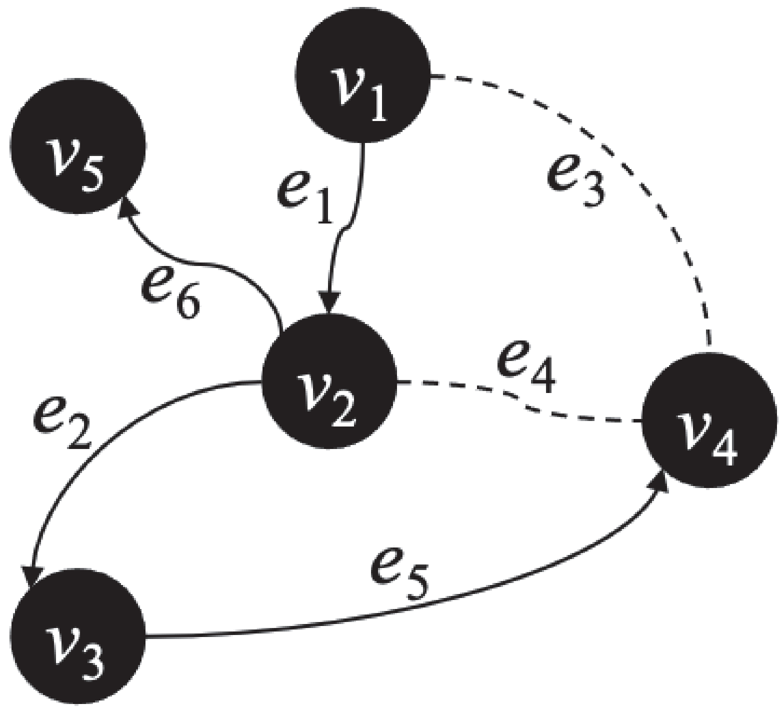
初值: V 中所有顶点的 visited 初值为 false

u.visited $\leftarrow$ true;

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

  if v.visited = false **then**

    DFS(G, v);



连通和DFS

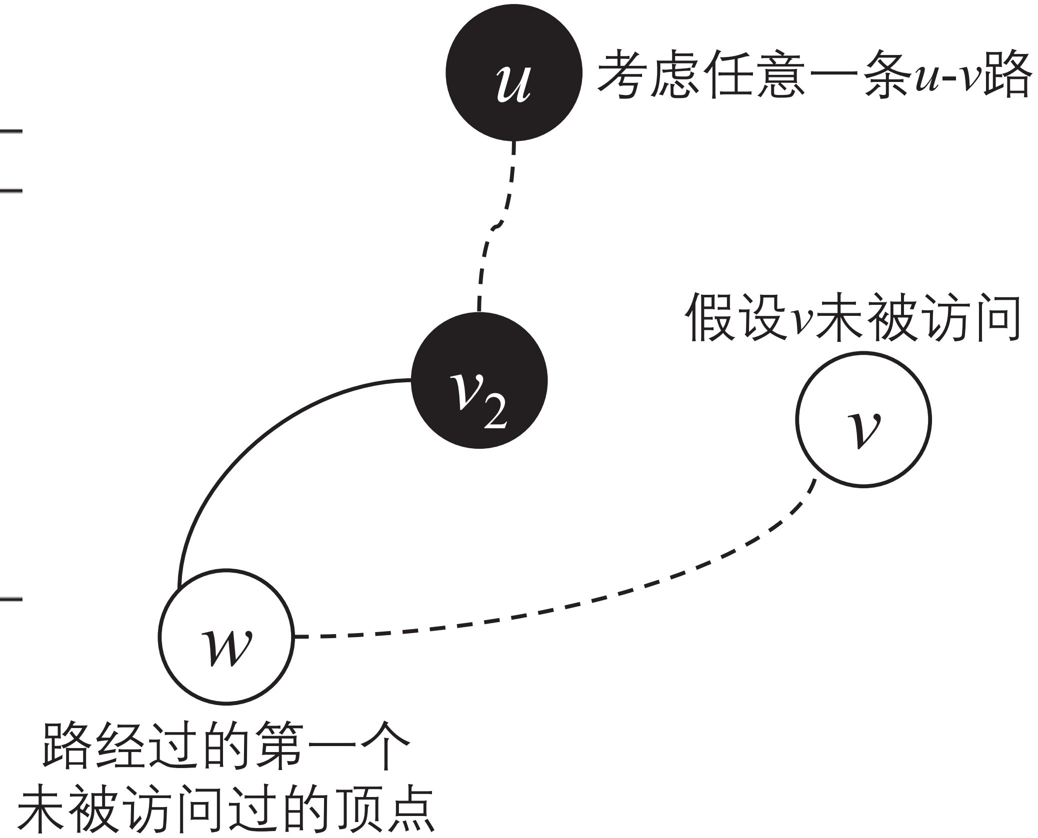
- 从顶点u出发运行DFS算法，为什么恰能访问与u连通的所有顶点？
  - 为什么连通的都能访问？
  - 为什么访问的都连通？

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

初值: V 中所有顶点的 visited 初值为 false

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        DFS(G, v);
```



连通和DFS

- 时间复杂度：$O(n + m)$

算法 2.1: DFS

输入: 图 $G = \langle V, E \rangle$，顶点 u

初值: V 中所有顶点的 visited 初值为 false

```
u.visited ← true;
foreach (u, v) ∈ E do
    if v.visited = false then
        DFS(G, v);
```