

玩儿转图论算法

liuyubobobo

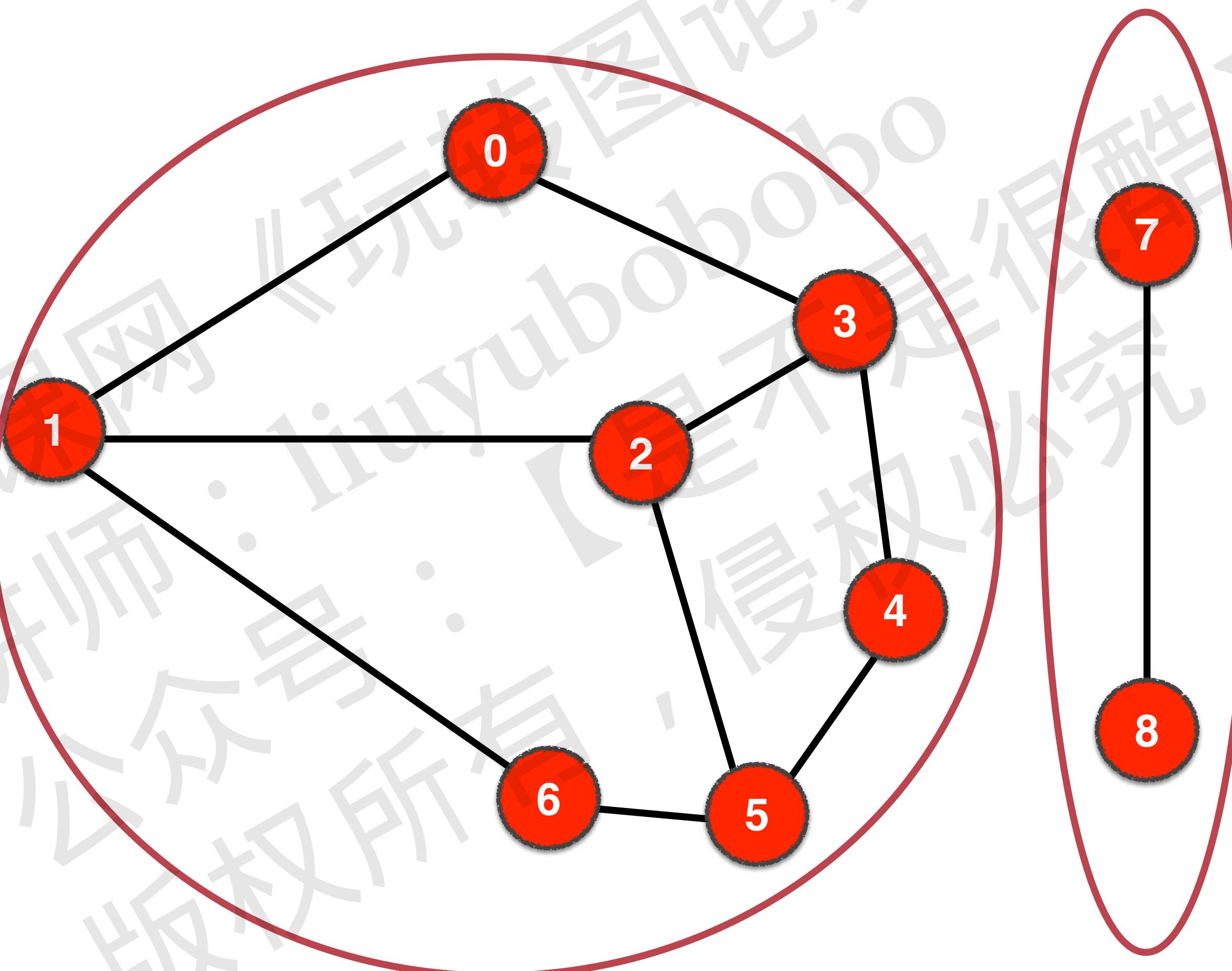
深度优先遍历的应用

liuyubobobo

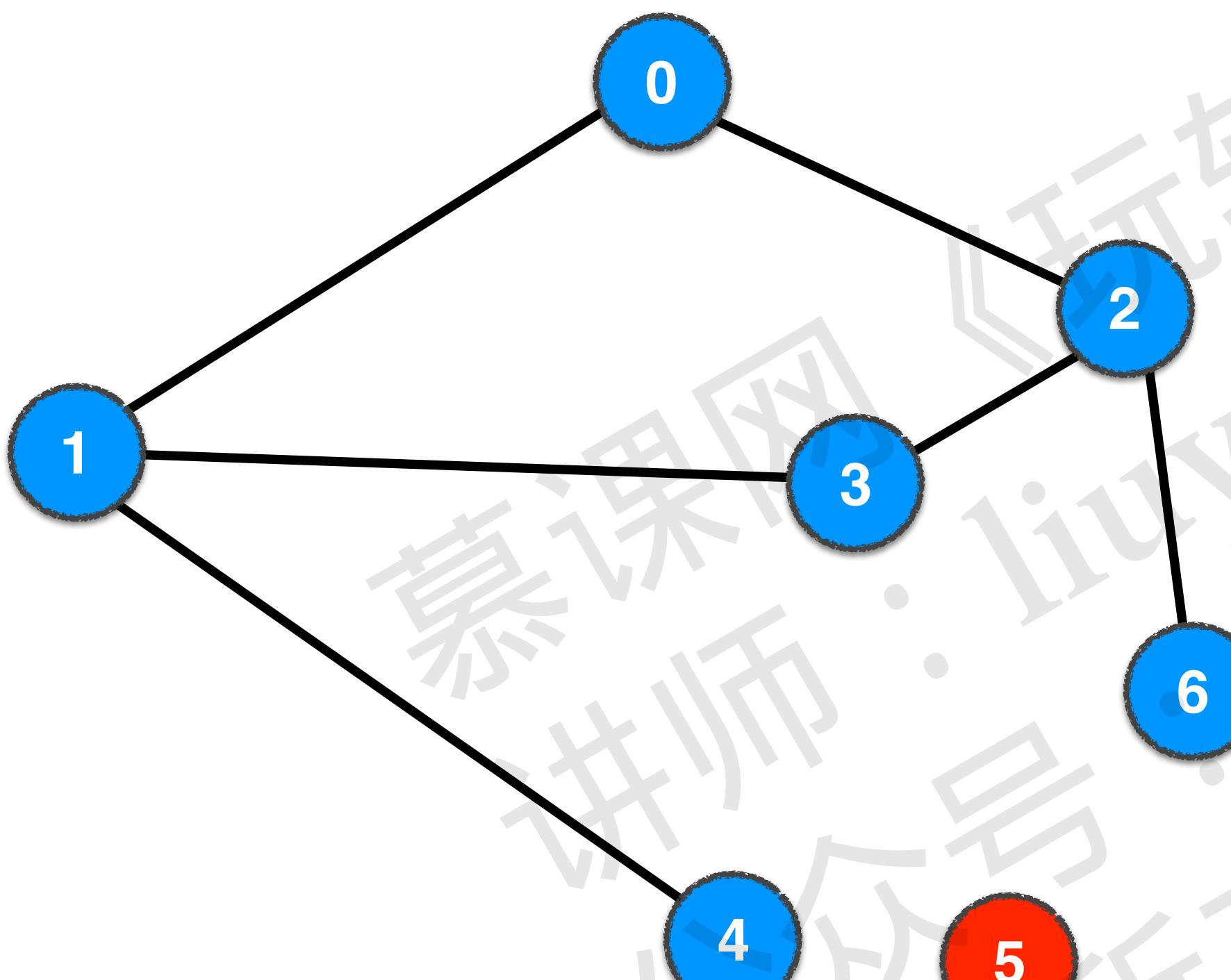
无向图的联通分量个数

liuyubobobo

无向图的联通分量

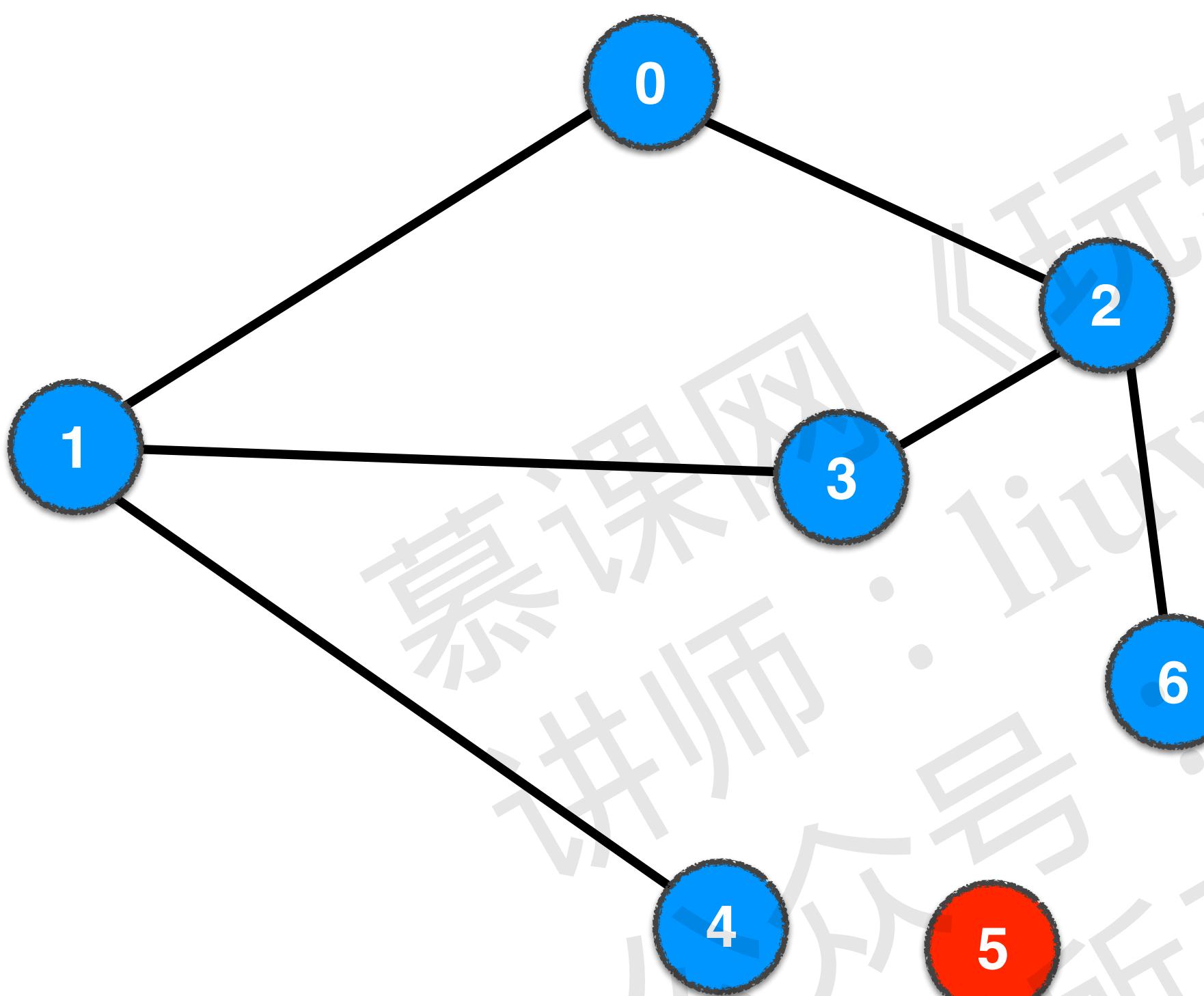


无向图的联通分量个数



```
visited[0...V] = false;  
  
for(int v = 0; v < V; v ++)  
if(!visited[v])  
dfs(v);  
  
dfs(int v){  
visited[w] = true;  
list.add(v);  
for(int w: adj(v))  
if(!visited[w])  
dfs(w);  
}
```

无向图的联通分量个数



联通分量
Connected Component

```
visited[0...V] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v);  
        cccout ++  
  
    }  
dfs(int v){  
  
    visited[w] = true;  
    list.add(v);  
  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w);  
  
    }
```

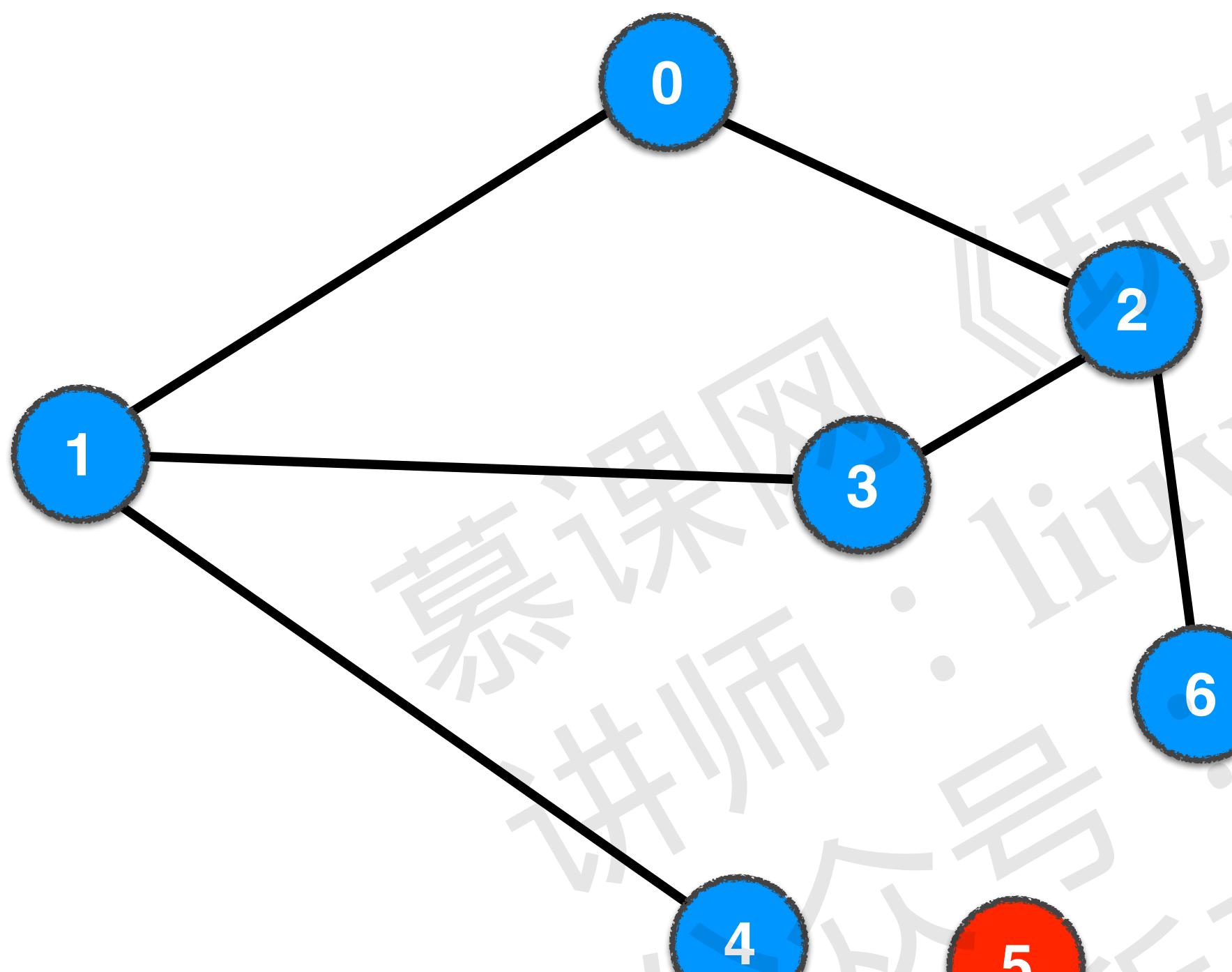
编程实践：实现联通分量算法类

慕课网 · liuyanbo · 《玩转图论算法》

具体求解无向图的联通分量

liuyubobobo

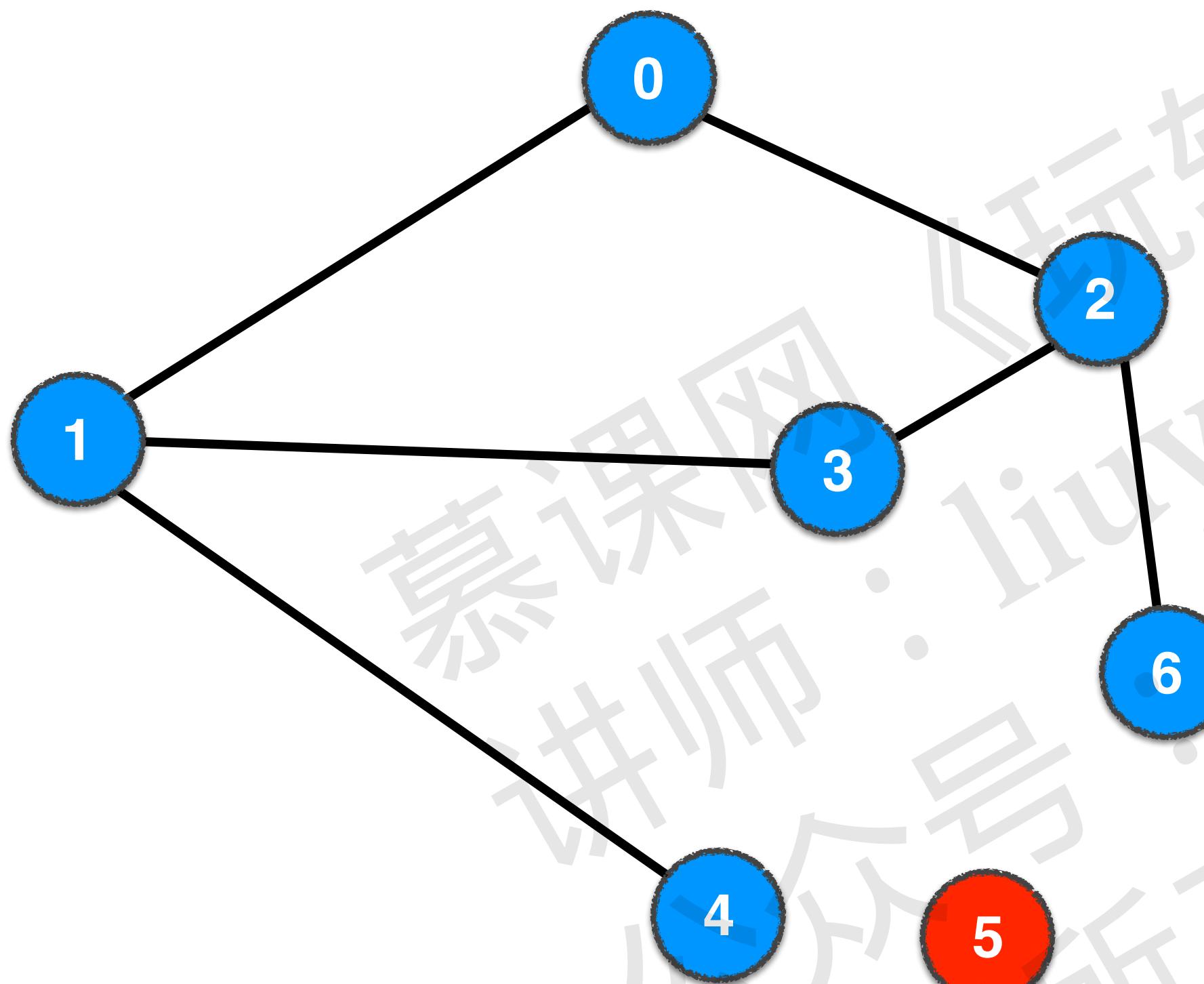
具体求解无向图的联通分量



联通分量
Connected Component

```
visited[0...V] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v);  
        cccout ++  
  
dfs(int v){  
  
    visited[w] = true;  
    list.add(v);  
  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w);  
}
```

具体求解无向图的联通分量



联通分量
Connected Component

```
visited[0...V] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v, list[i]);  
    cccout ++;  
    i ++;  
  
}  
  
dfs(int v, list){  
  
    visited[w] = true;  
    list.add(v);  
  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w, list);  
  
}
```

具体求解无向图的联通分量

```
visited[0...V] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v);  
        cccout ++  
}  
  
dfs(int v){  
    visited[w] = true;  
  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w);  
}  
}
```

0	1	2	3	4	5	6	
visited	F	F	F	F	F	F	
visited	T	T	T	T	T	F	T
visited	T	T	T	T	T	T	T
0	1	2	3	4	5	6	
visited	-1	-1	-1	-1	-1	-1	-1
visited	0	0	0	0	0	-1	0
visited	0	0	0	0	0	1	0

慕课网 编程实践：修改visited

慕课网《玩转图论算法》
讲师：liuyibobobo
公众号：【是极客必修】
版权所有，侵权必究

编程实践：查看两点间是否联通

慕课网 · liuyan · 《图论与算法》 · 课件 · 作者 · bobobo · 很酷

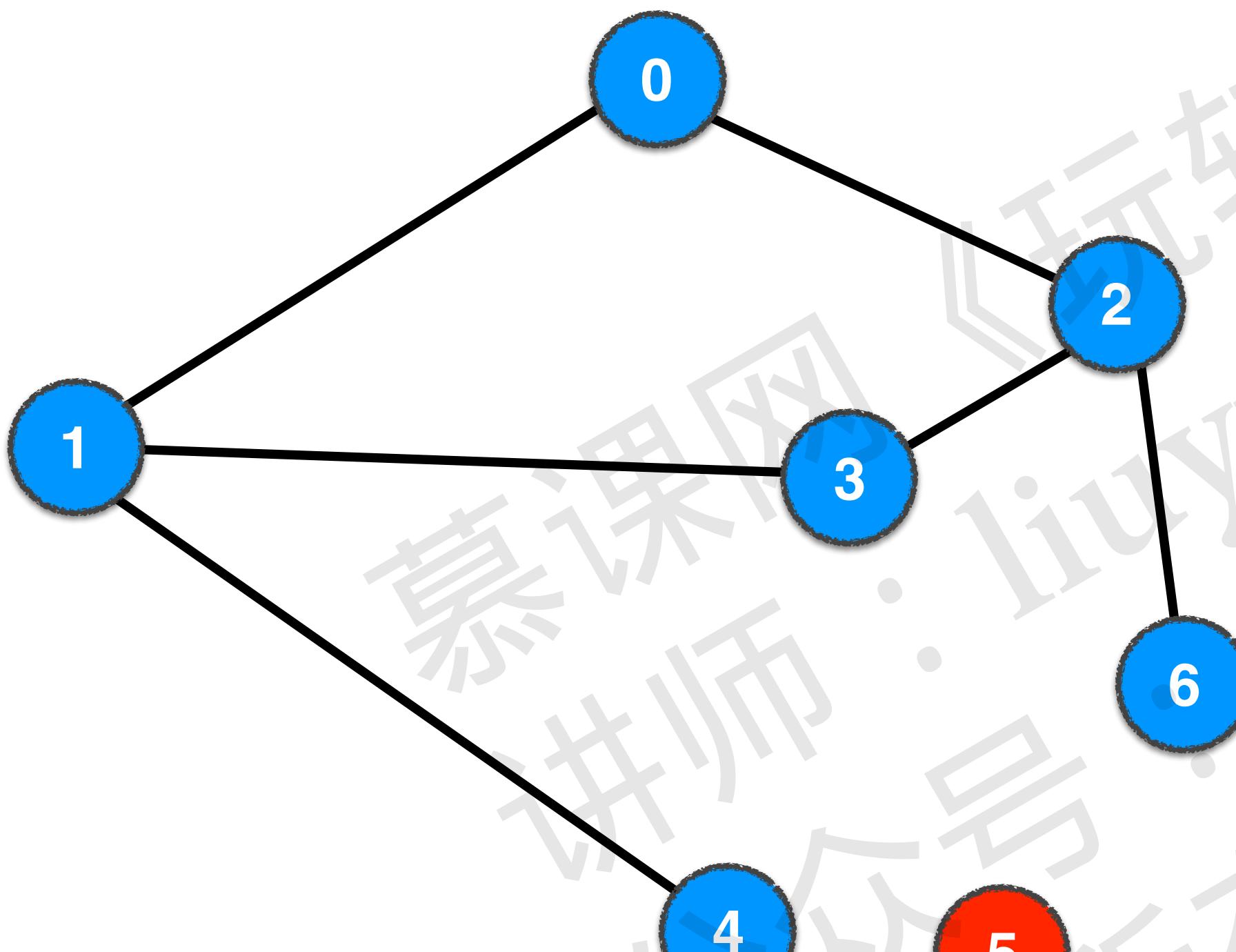
编程实践：求解具体的联通分量

慕课网 · liuyan · 《玩转图论算法》

路径问题

liuyubobobo

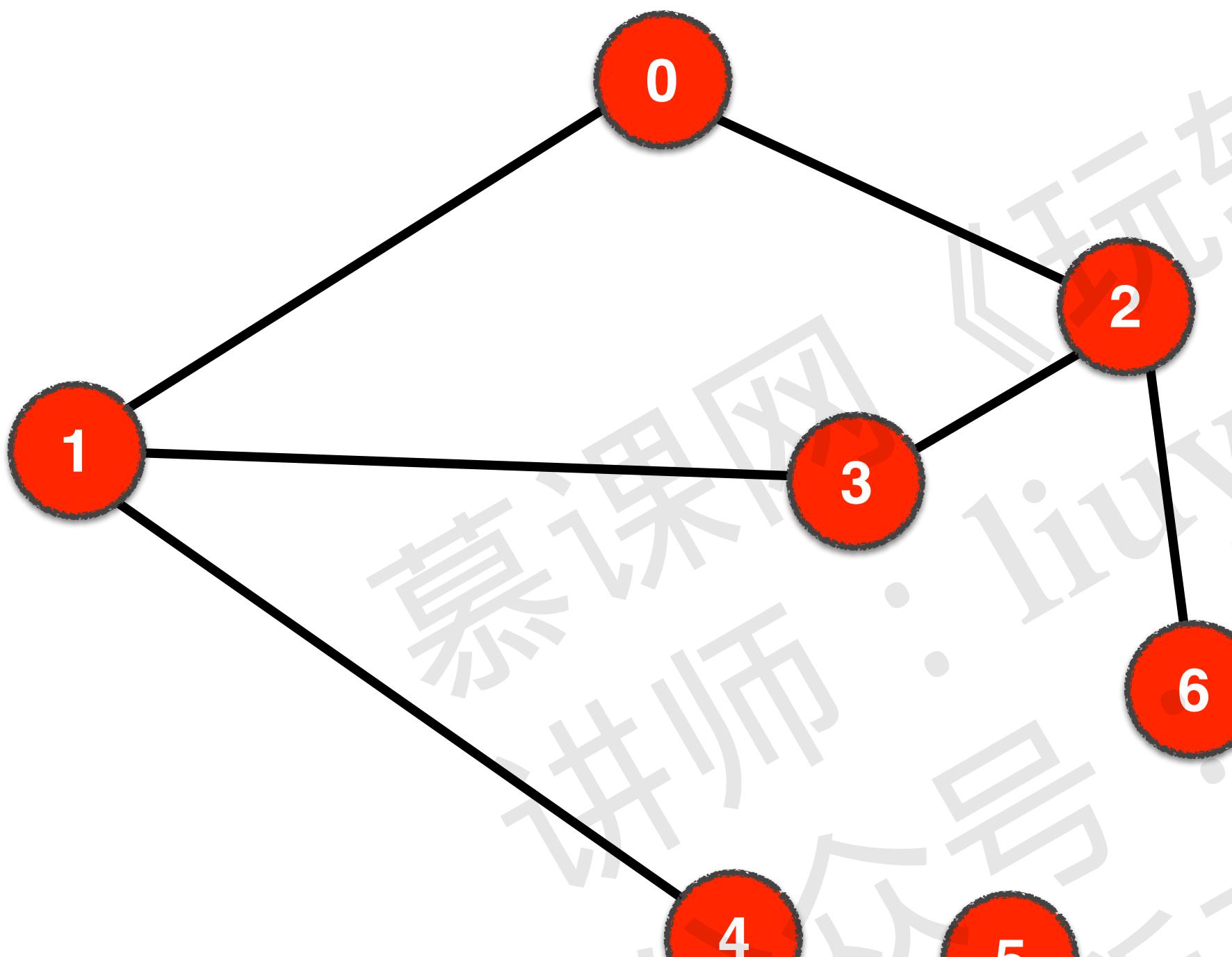
路径问题



两点在同一个联通分量，
意味着两点间有路径

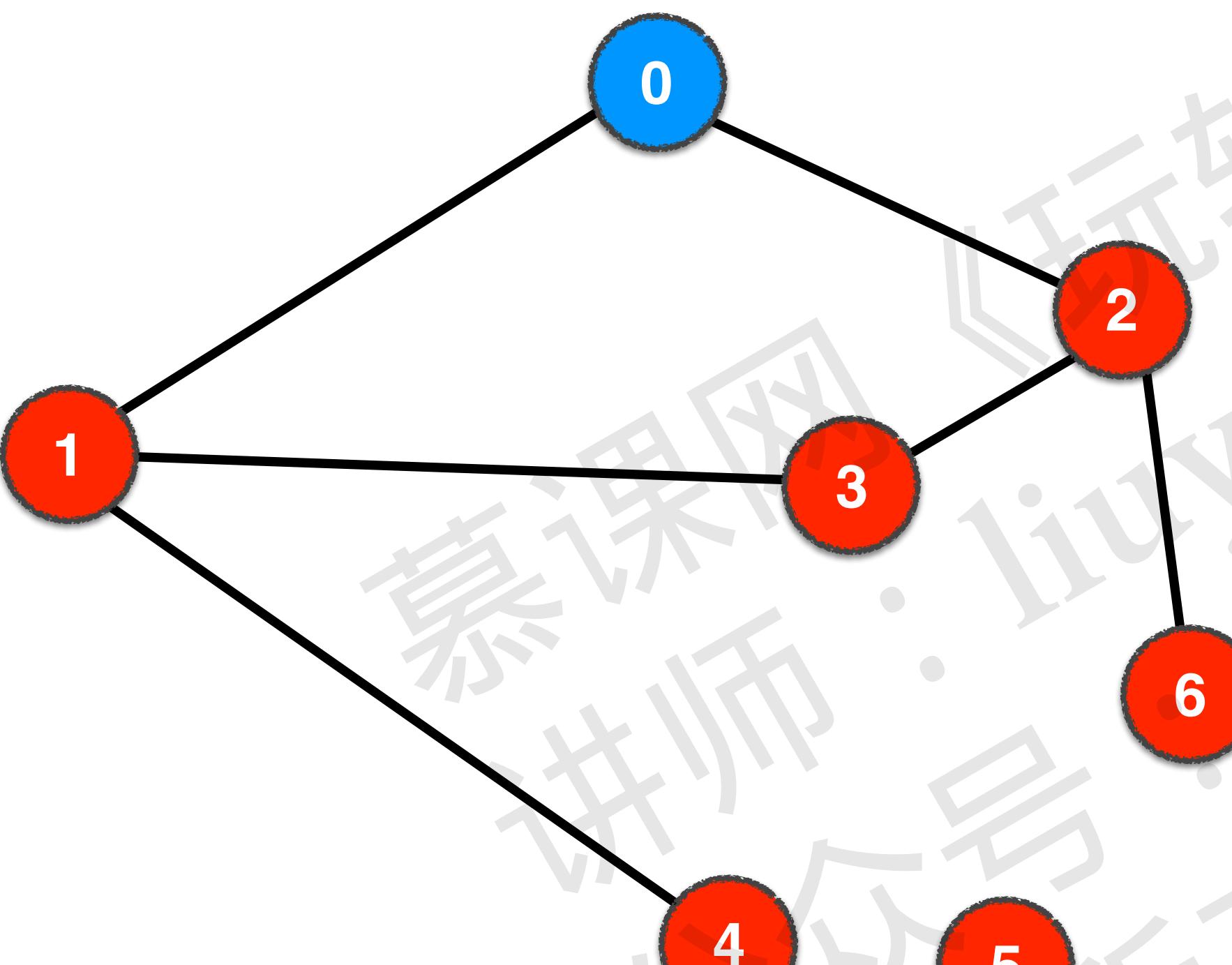
路径怎么求？

路径问题



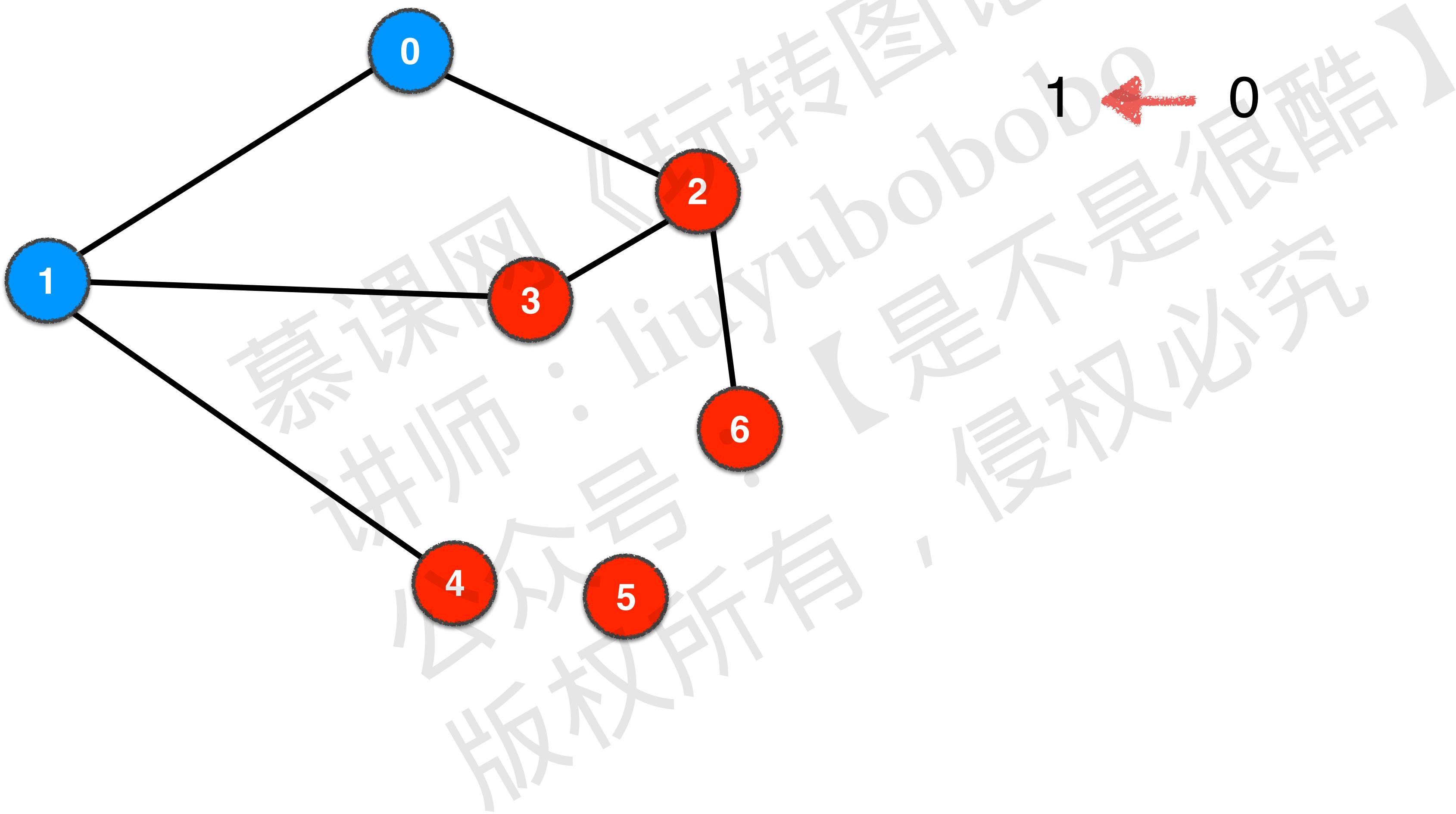
慕课网 · liuyubobobo · 《图论与算法》 · 路径问题 · 版权所有，侵权必究

路径问题

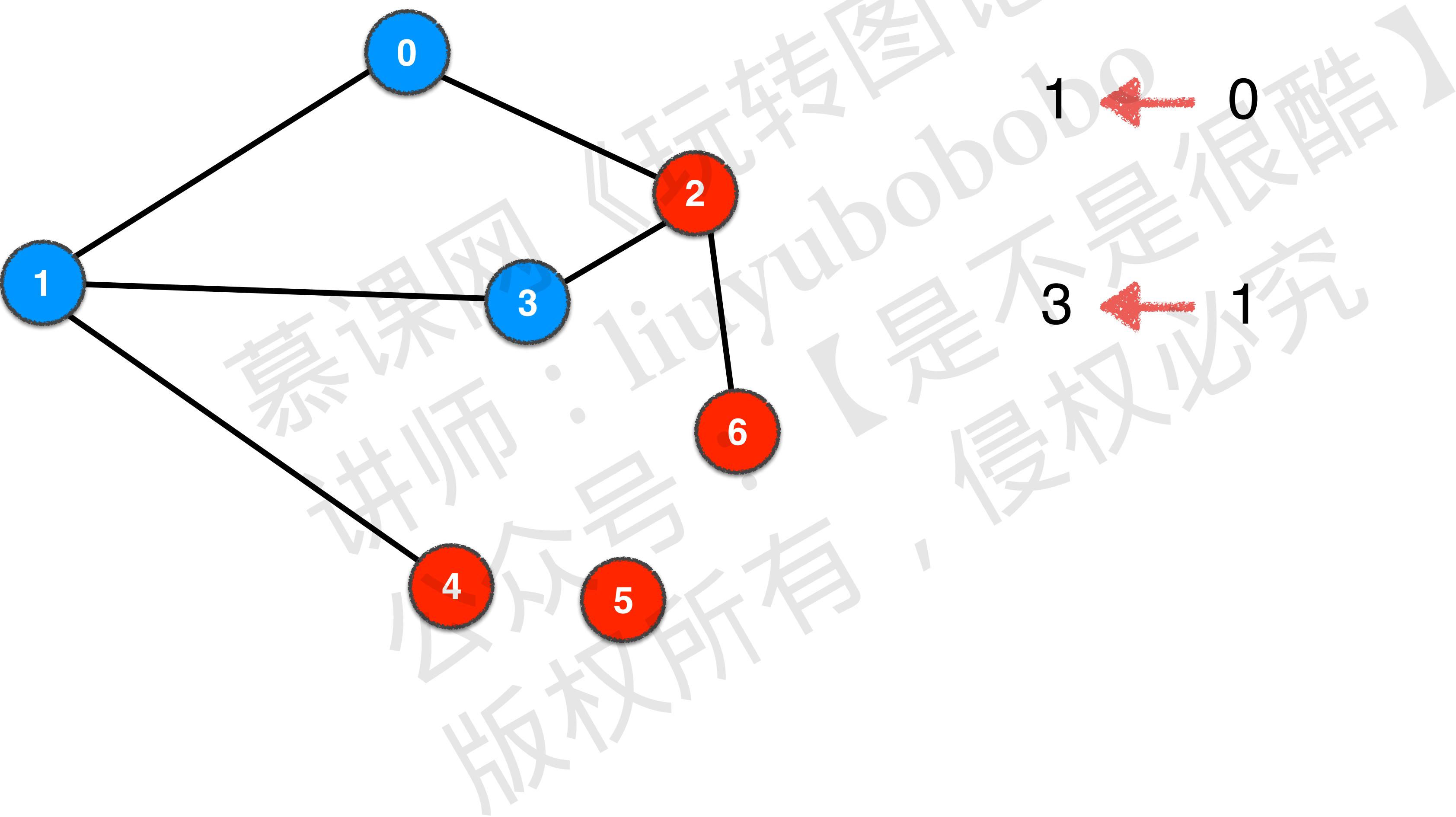


慕课网 · liuyubobobo · 《图论与算法》 · 路径问题 · 版权所有，侵权必究

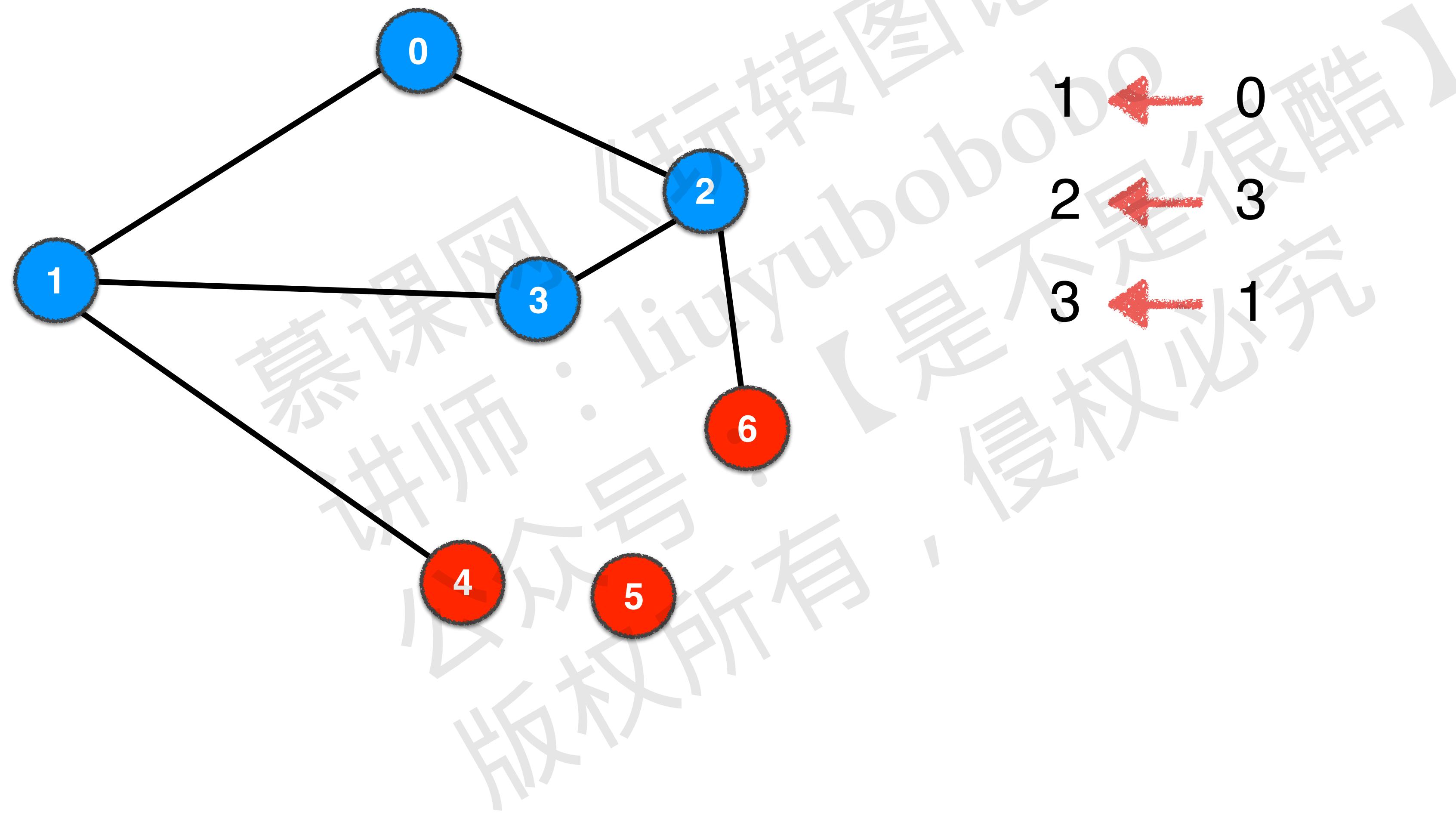
路径问题



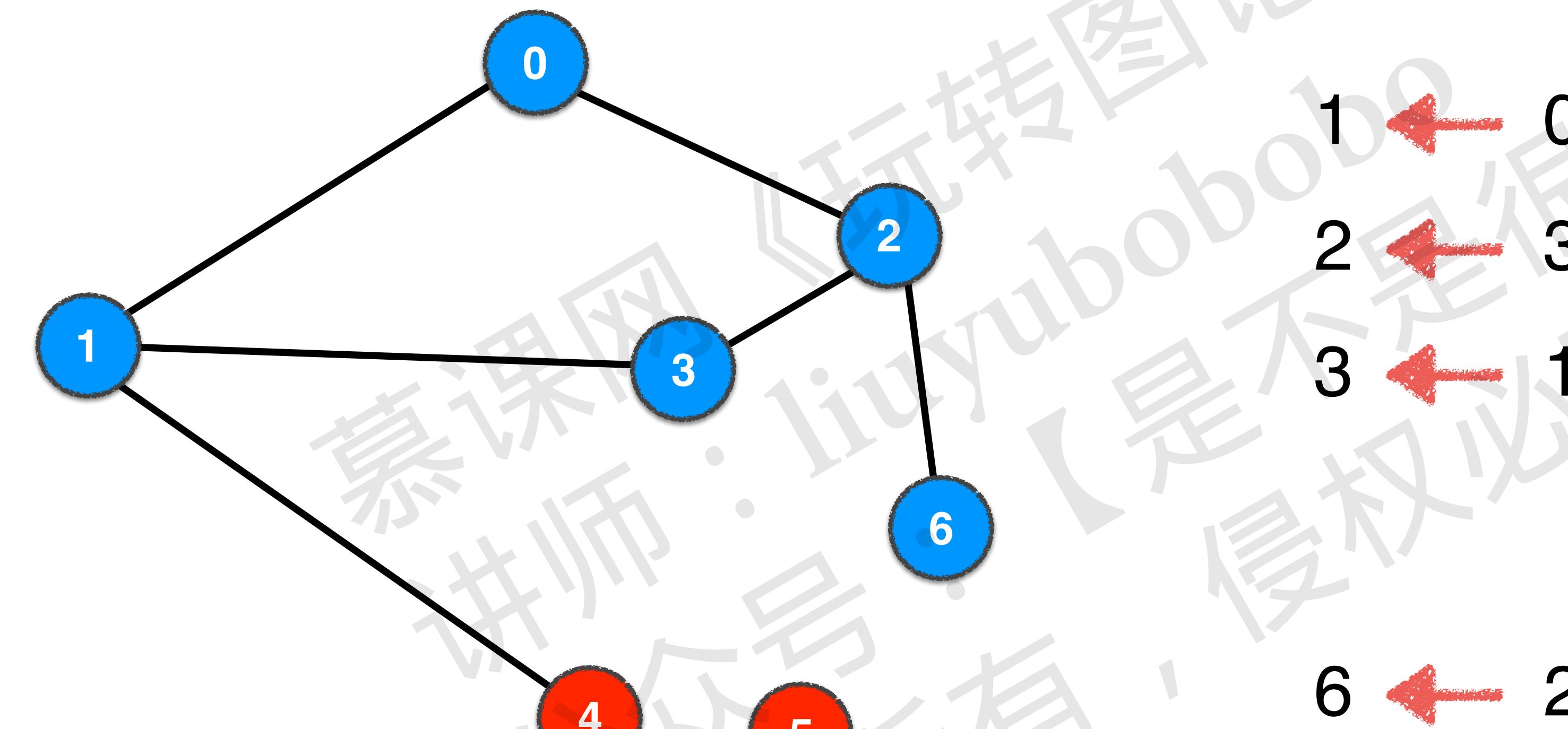
路径问题



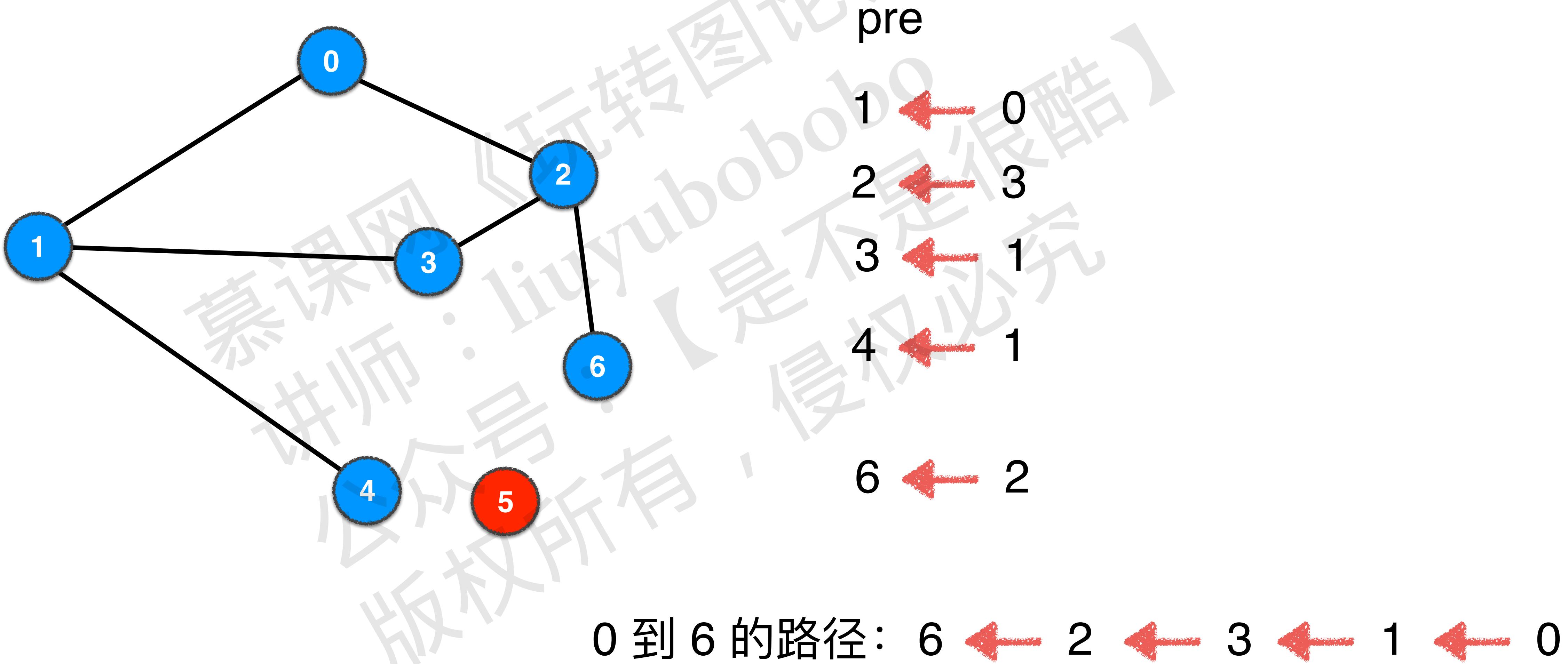
路径问题



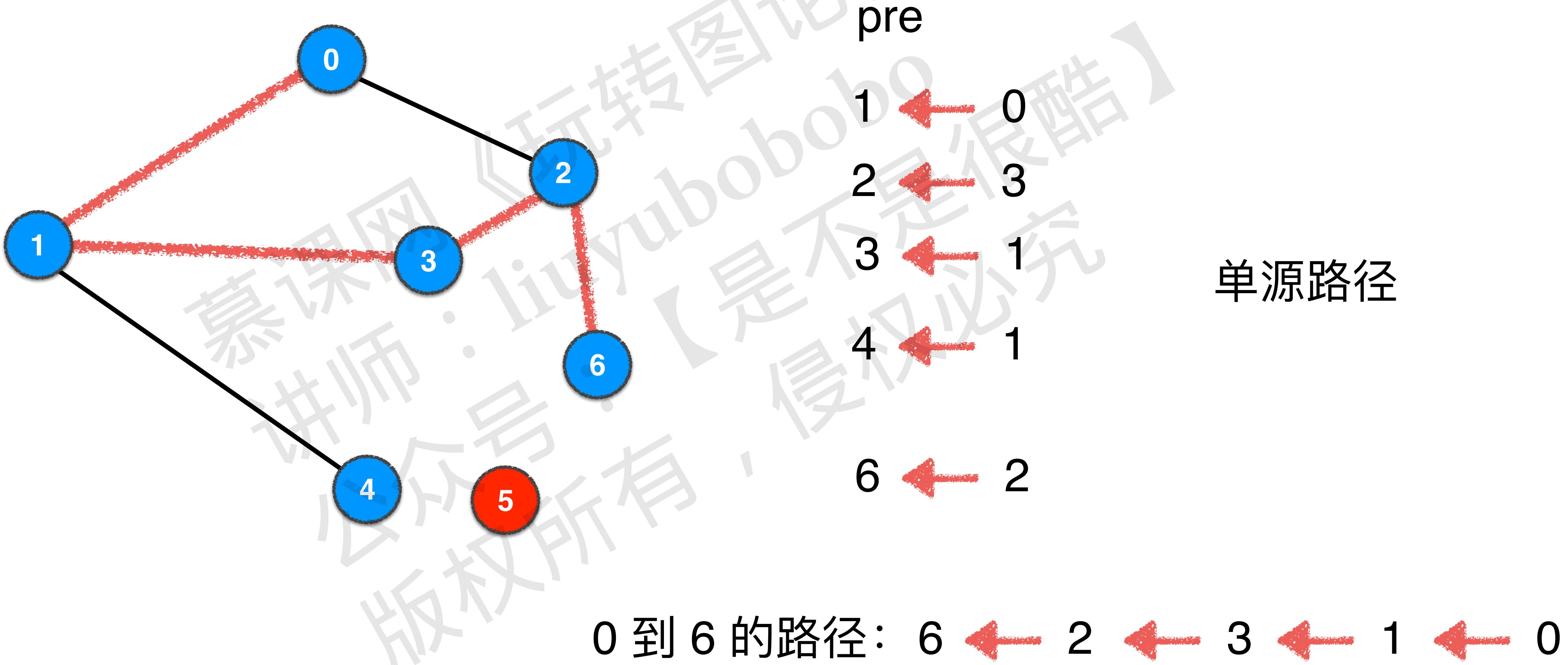
路径问题



路径問題



路径问题



编程实现路径问题

liuyubobobo

编程实践：求解单源路径问题

慕课网 · liuyibobobo · 《玩转图论算法》

路径问题的小优化

liuyubobobo

所有点对路径问题

liuyubobobo

编程实践：路径问题优化

慕课网 · lituyubobobo · 免费学习 · 每日必做
讲师 · 刘育波 · 《玩转图论算法》法
版权所有 · 公众号 · 版权所有

编程实践：所有点对路径问题

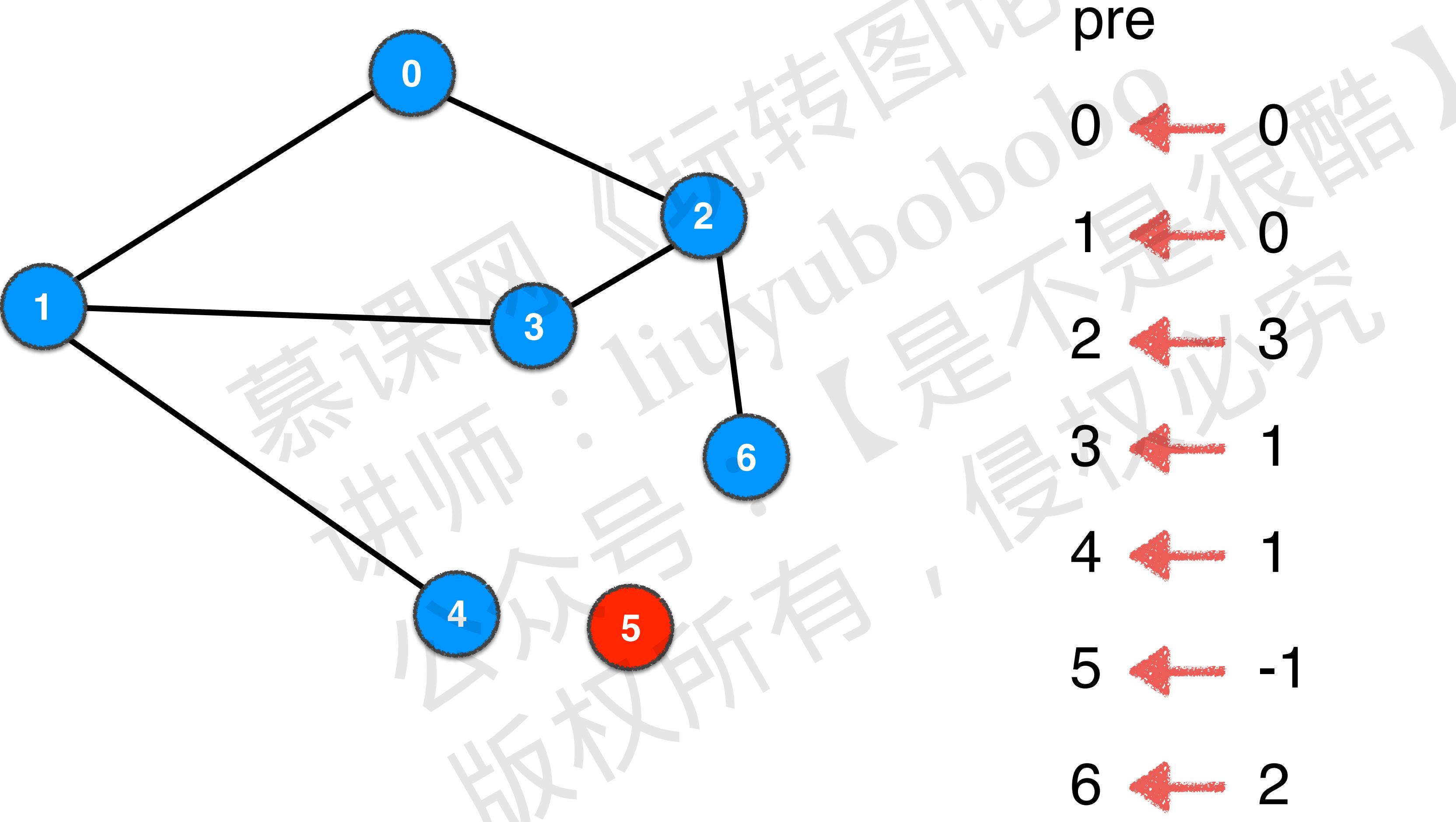
慕课网 · liuyibobobo · 《玩转图论算法》

提前结束递归

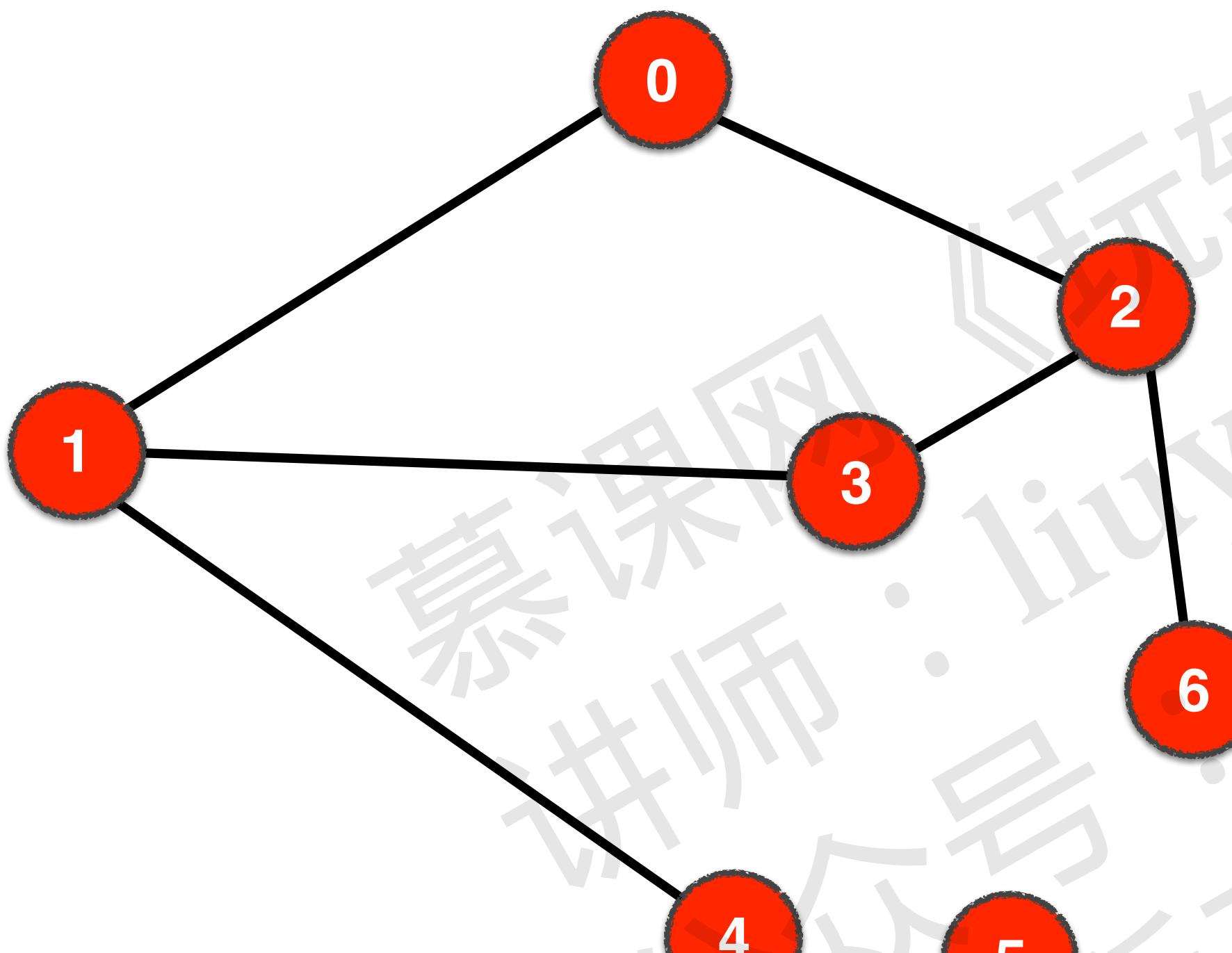
路径问题的另一个优化

liuyubobobo

路径问题

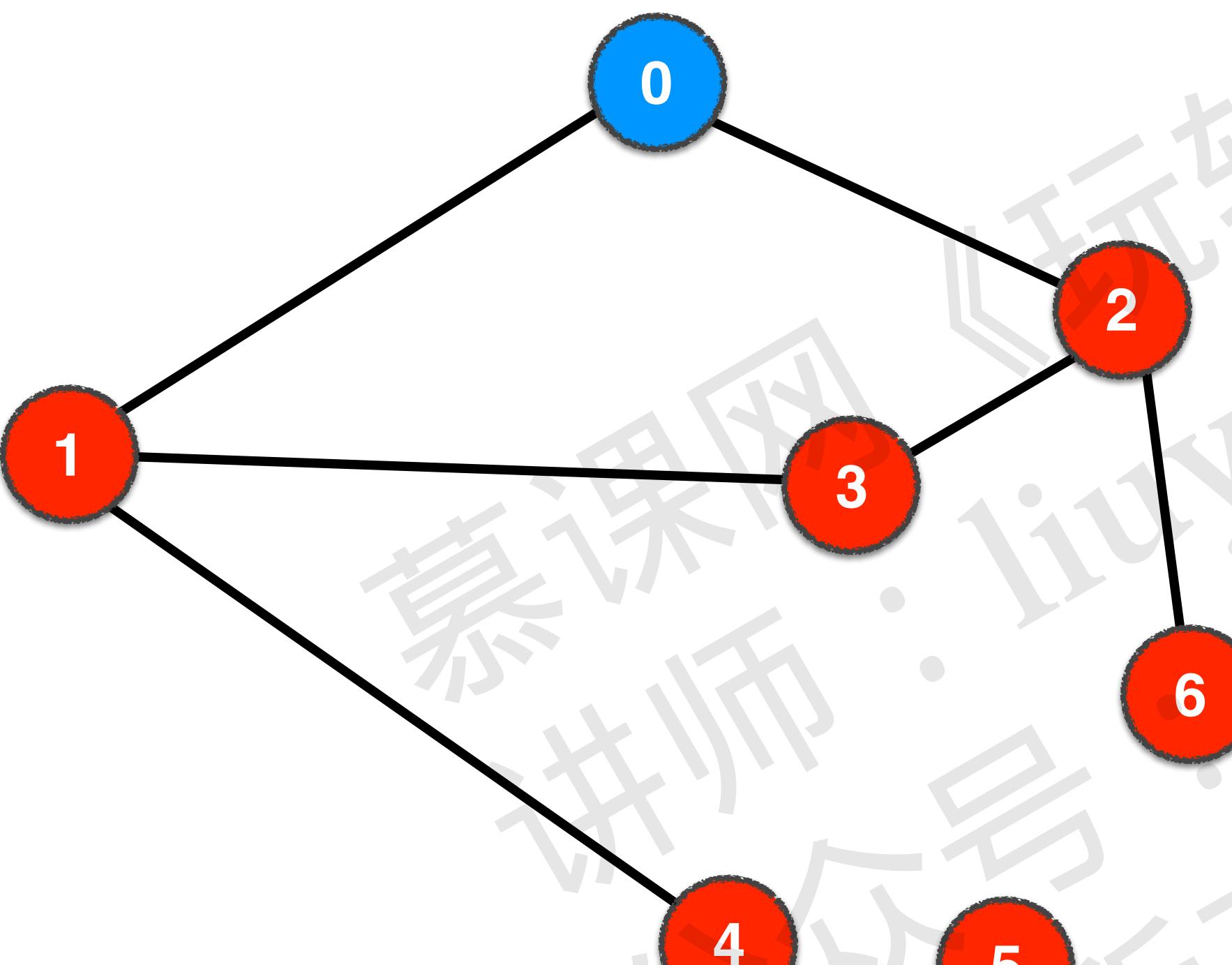


路径问题

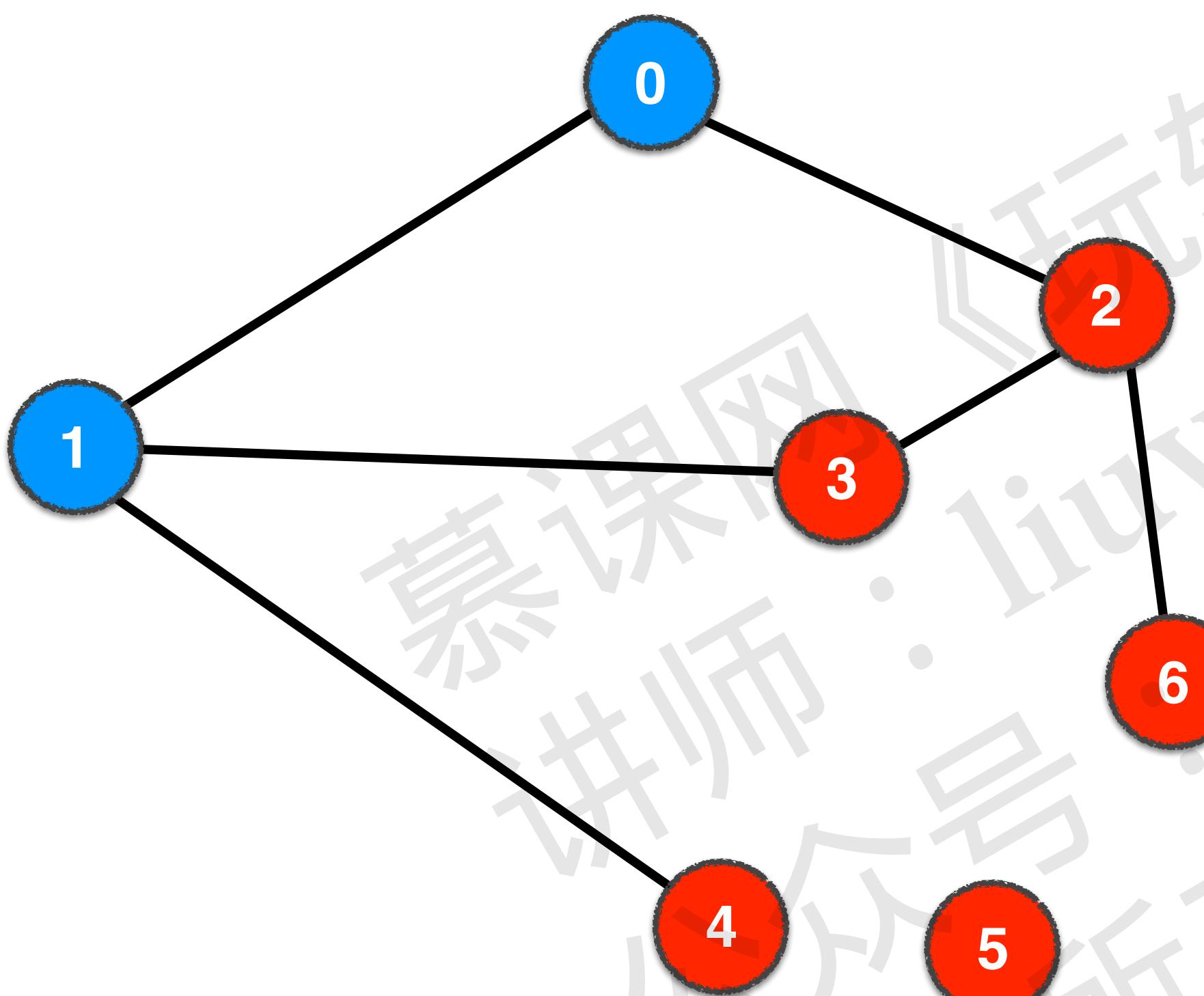


慕课网 · liuyubobobo · 《图论与算法》 · 路径问题 · 版权所有，侵权必究

路径问题

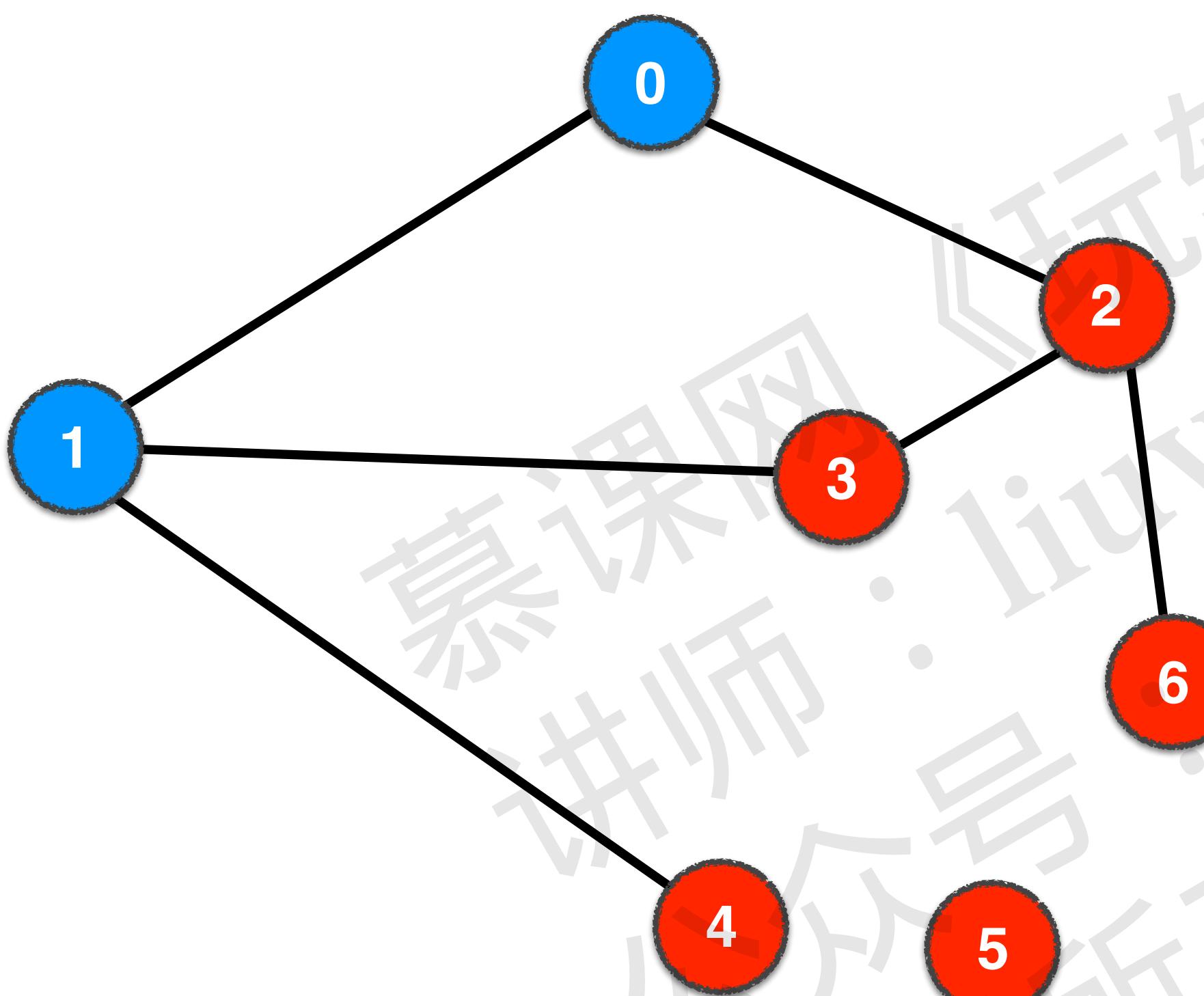


路径问题



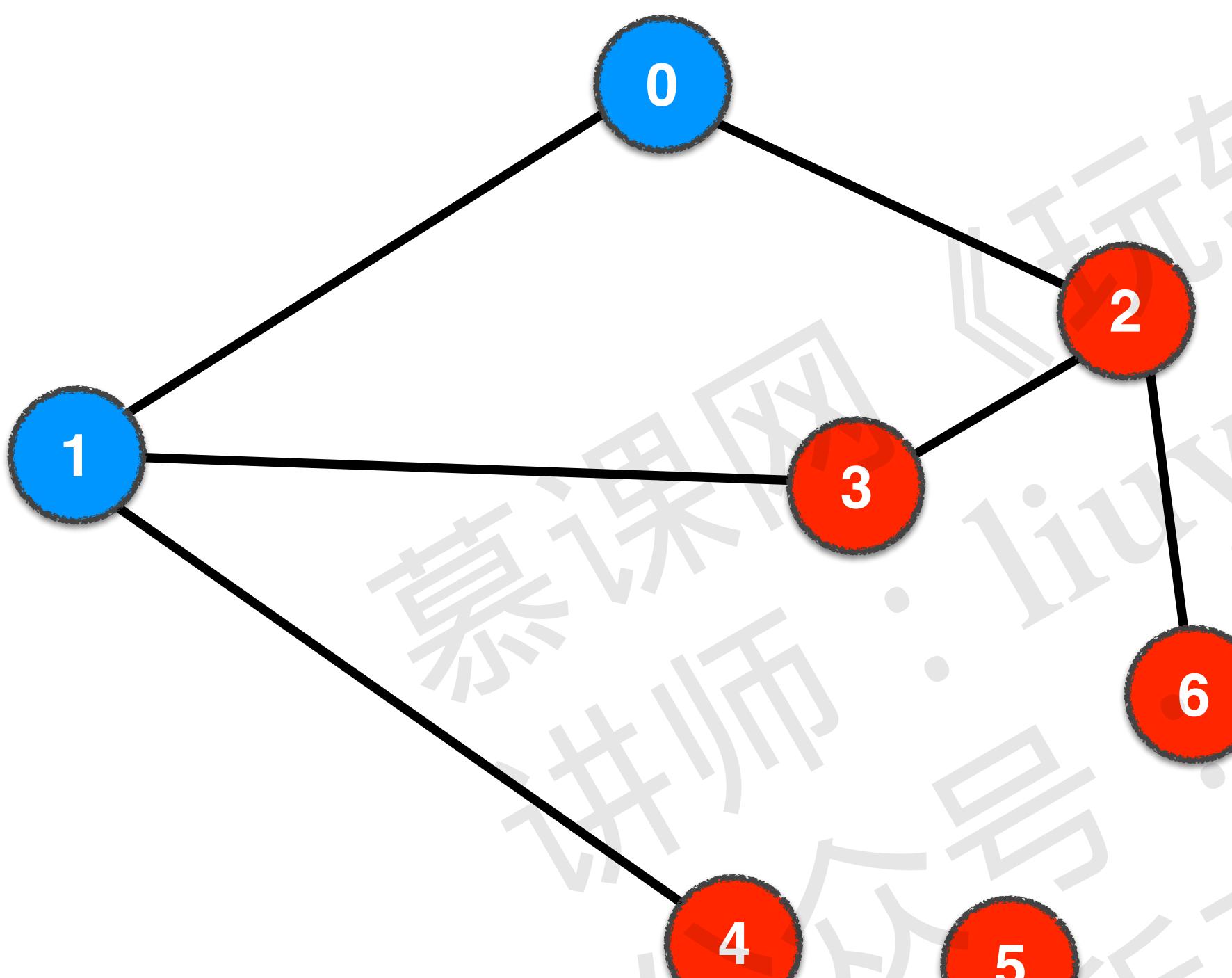
```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
dfs(s, s);  
  
dfs(int v, int parent){  
    visited[v] = true;  
    pre[v] = parent;  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w, v);  
}
```

路径问题



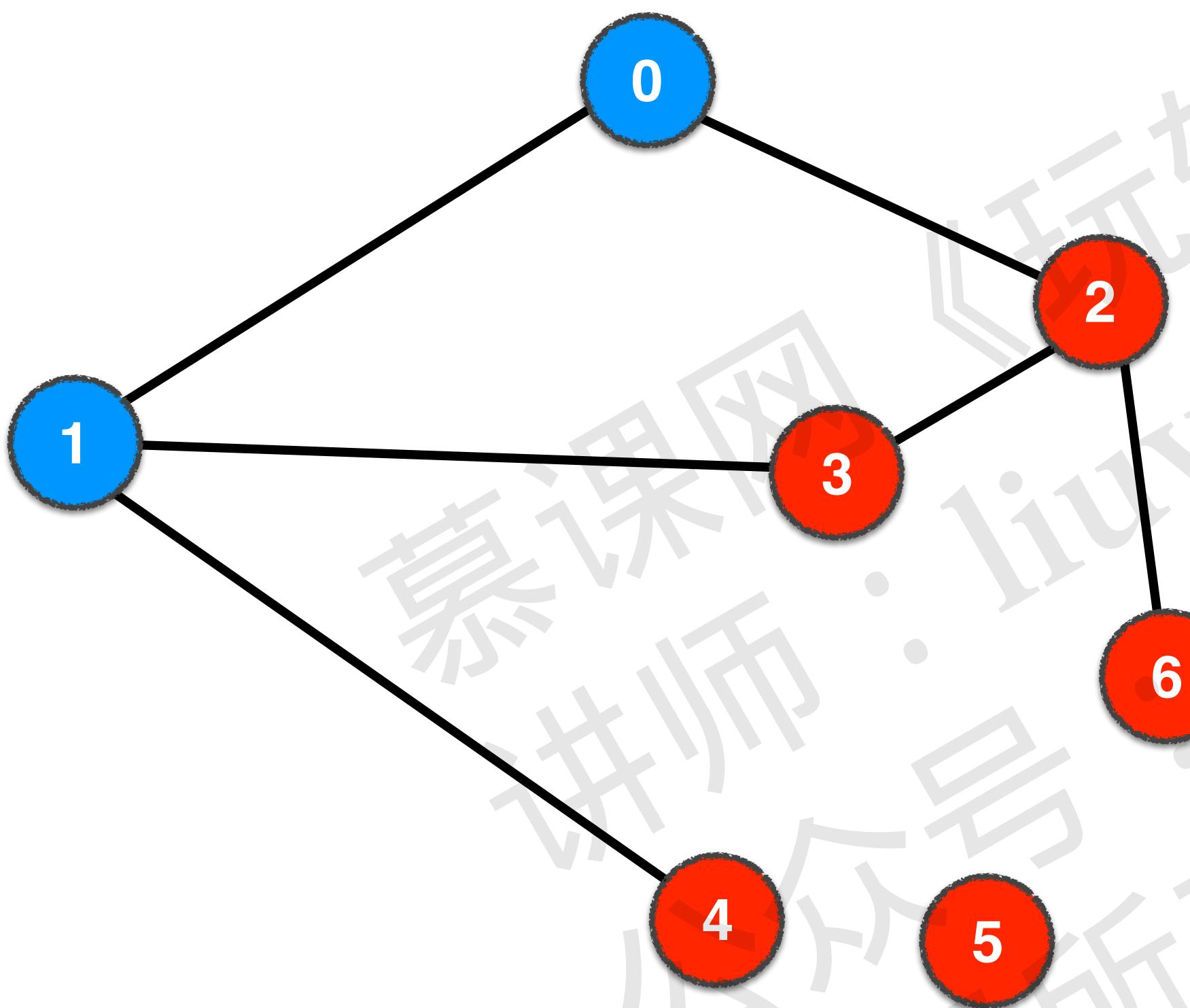
```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
dfs(s, s);  
  
dfs(int v, int parent){  
    visited[v] = true;  
    pre[v] = parent;  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w, v);  
}
```

路径问题



```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
dfs(s, s);  
  
dfs(int v, int parent){  
    visited[v] = true;  
    pre[v] = parent;  
    if(v == t) return true;  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w, v);  
}
```

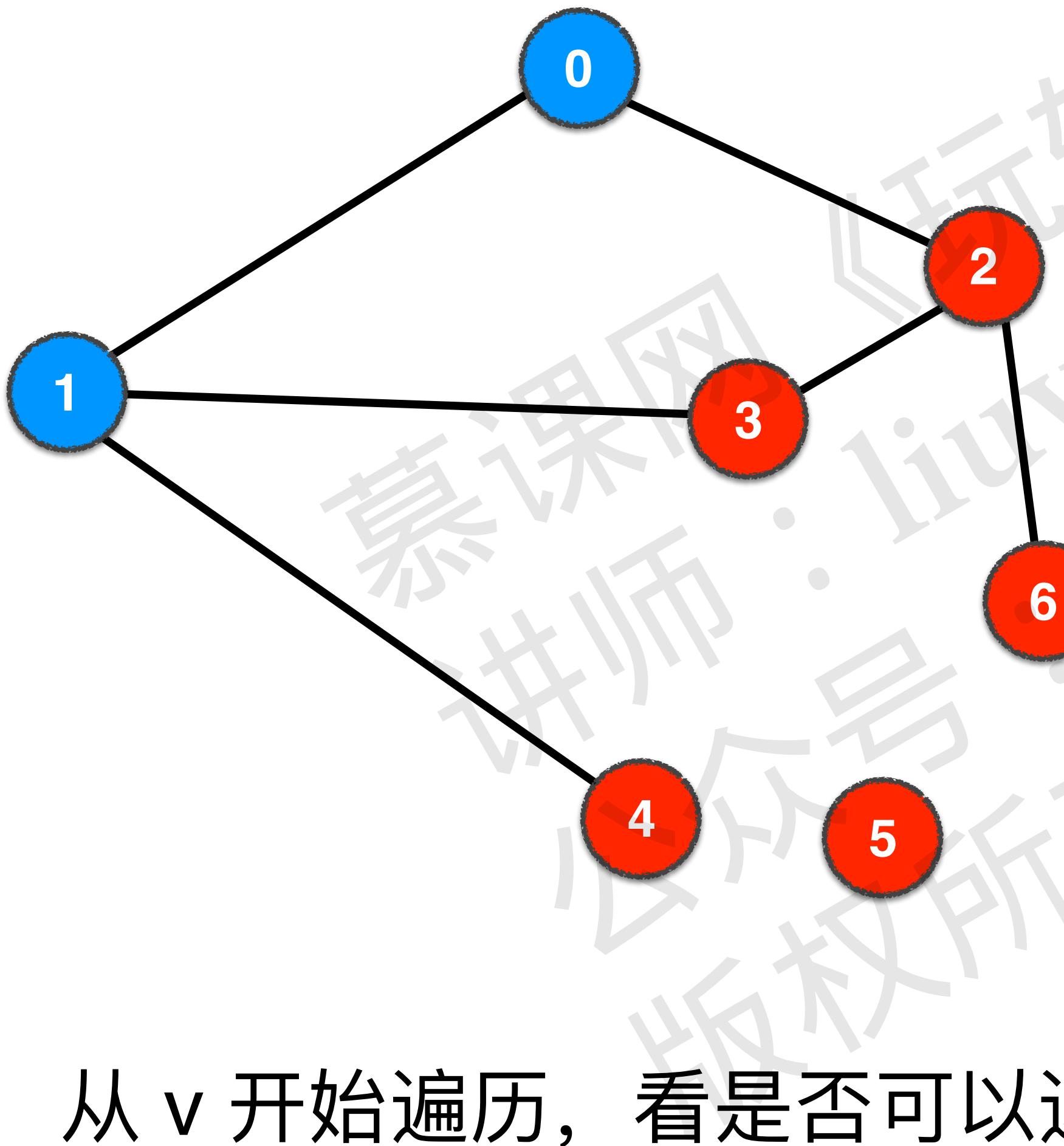
路径问题



从 v 开始遍历，看是否可以达到 t

```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
dfs(s, s);  
  
bool dfs(int v, int parent){  
    visited[v] = true;  
    pre[v] = parent;  
    if(v == t) return true;  
    for(int w: adj(v))  
        if(!visited[w])  
            dfs(w, v);  
}
```

路径问题



```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
dfs(s, s);  
  
bool dfs(int v, int parent){  
    visited[v] = true;  
    pre[v] = parent;  
    if(v == t) return true;  
    for(int w: adj(v))  
        if(!visited[w])  
            if(dfs(w, v)) return true;  
    return false;  
}
```

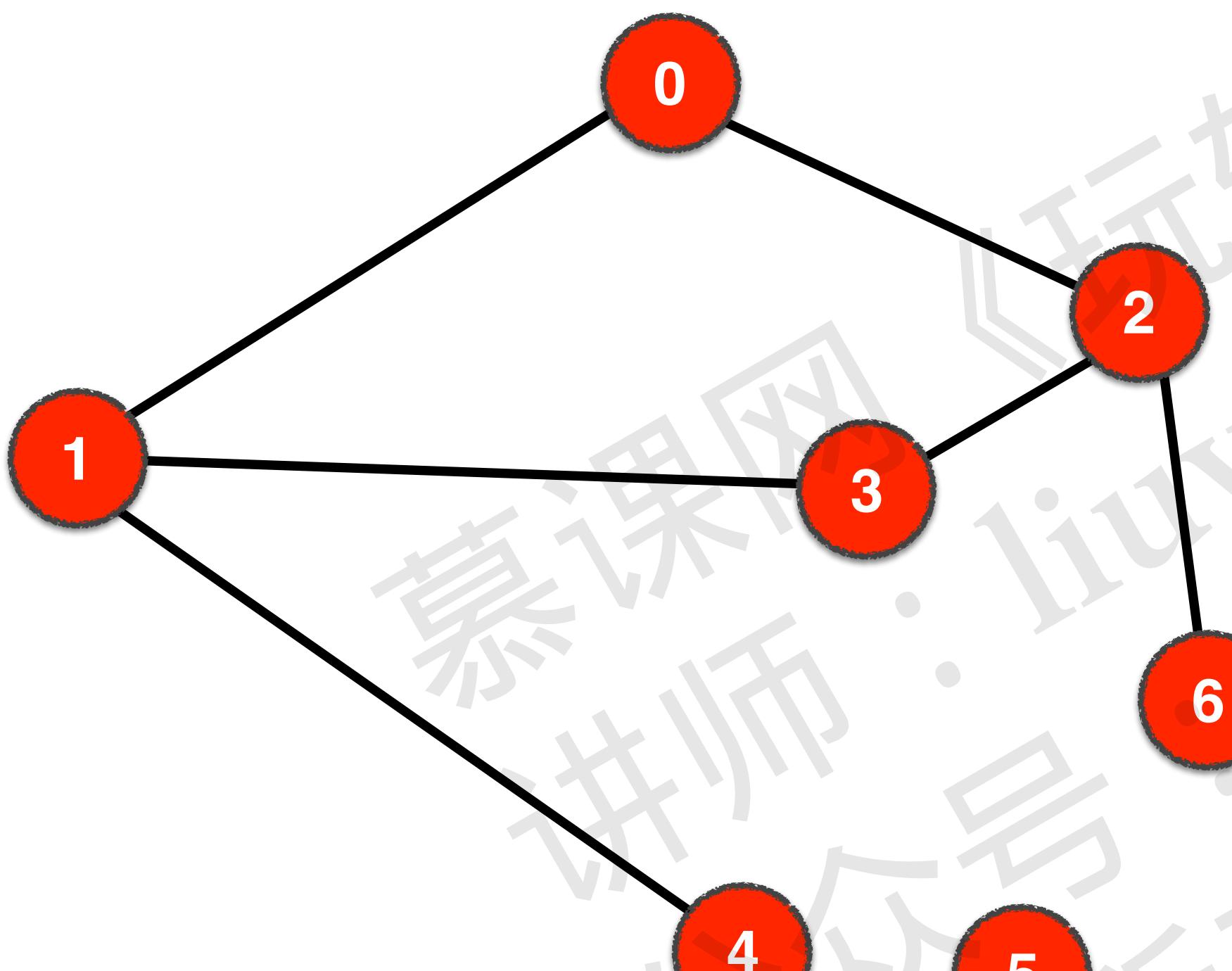
编程实践：两点间路径问题优化

慕课网 · liuyanbo · 《玩转图论算法》

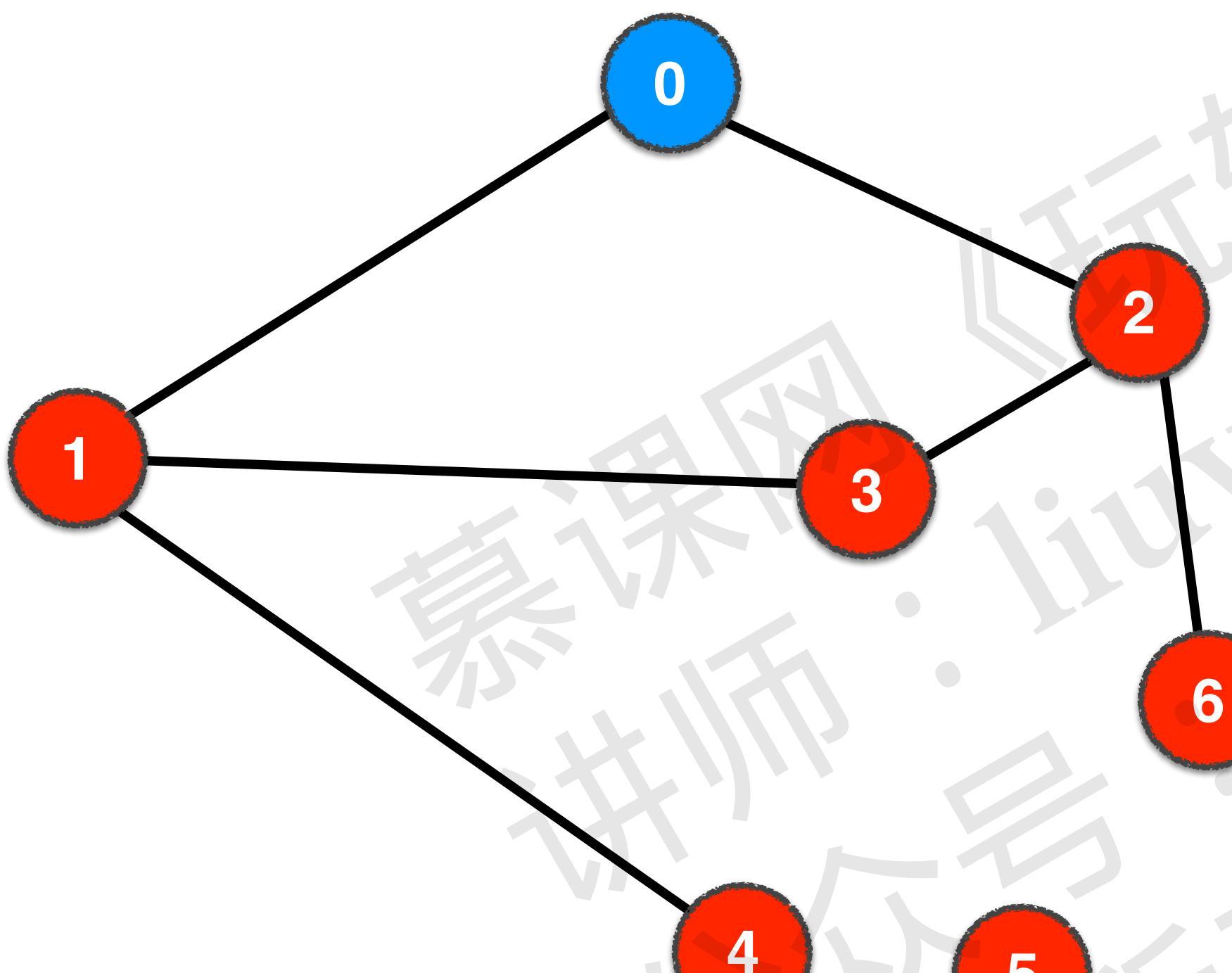
检测无向图中的环

liuyubobobo

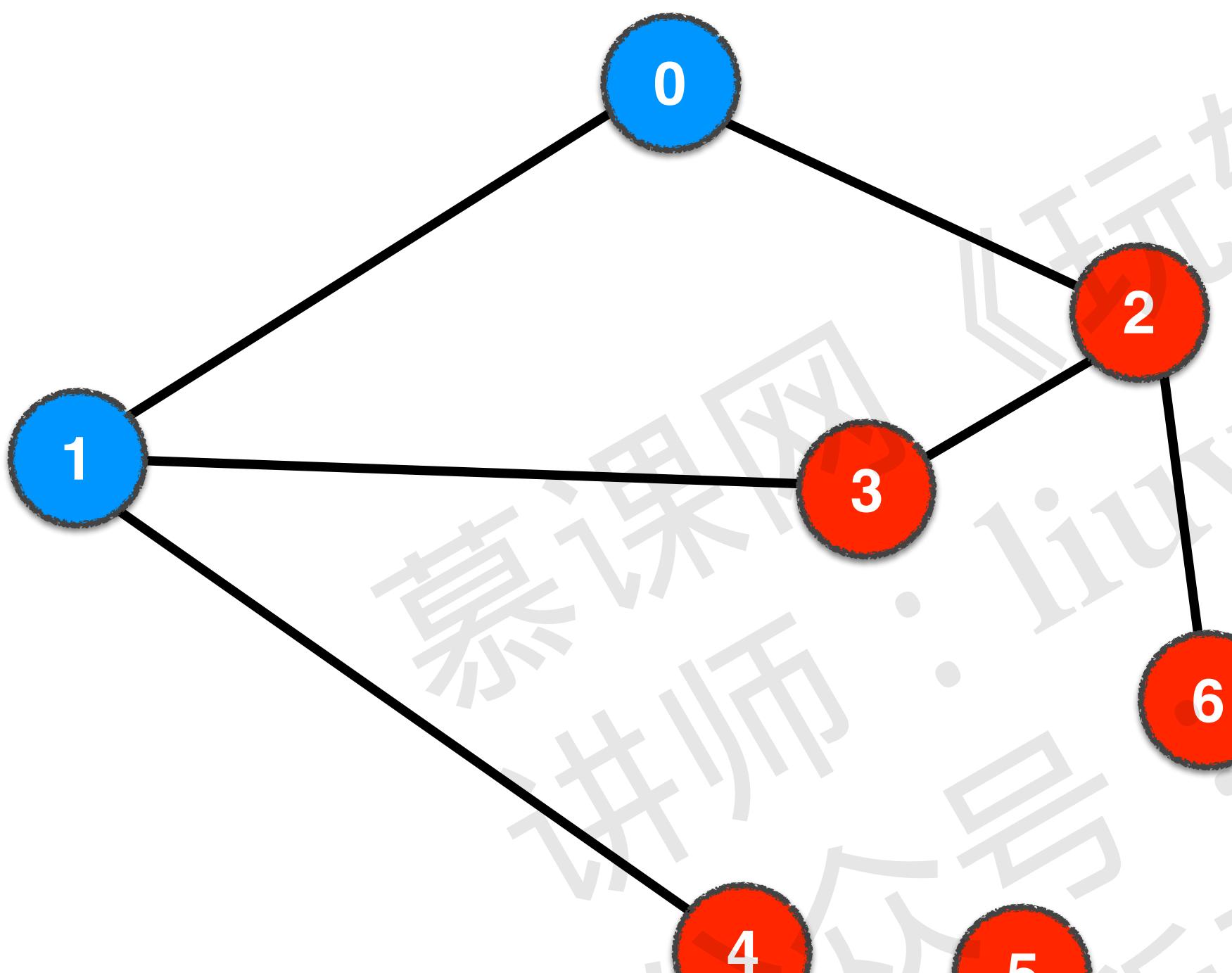
检测无向图中的环



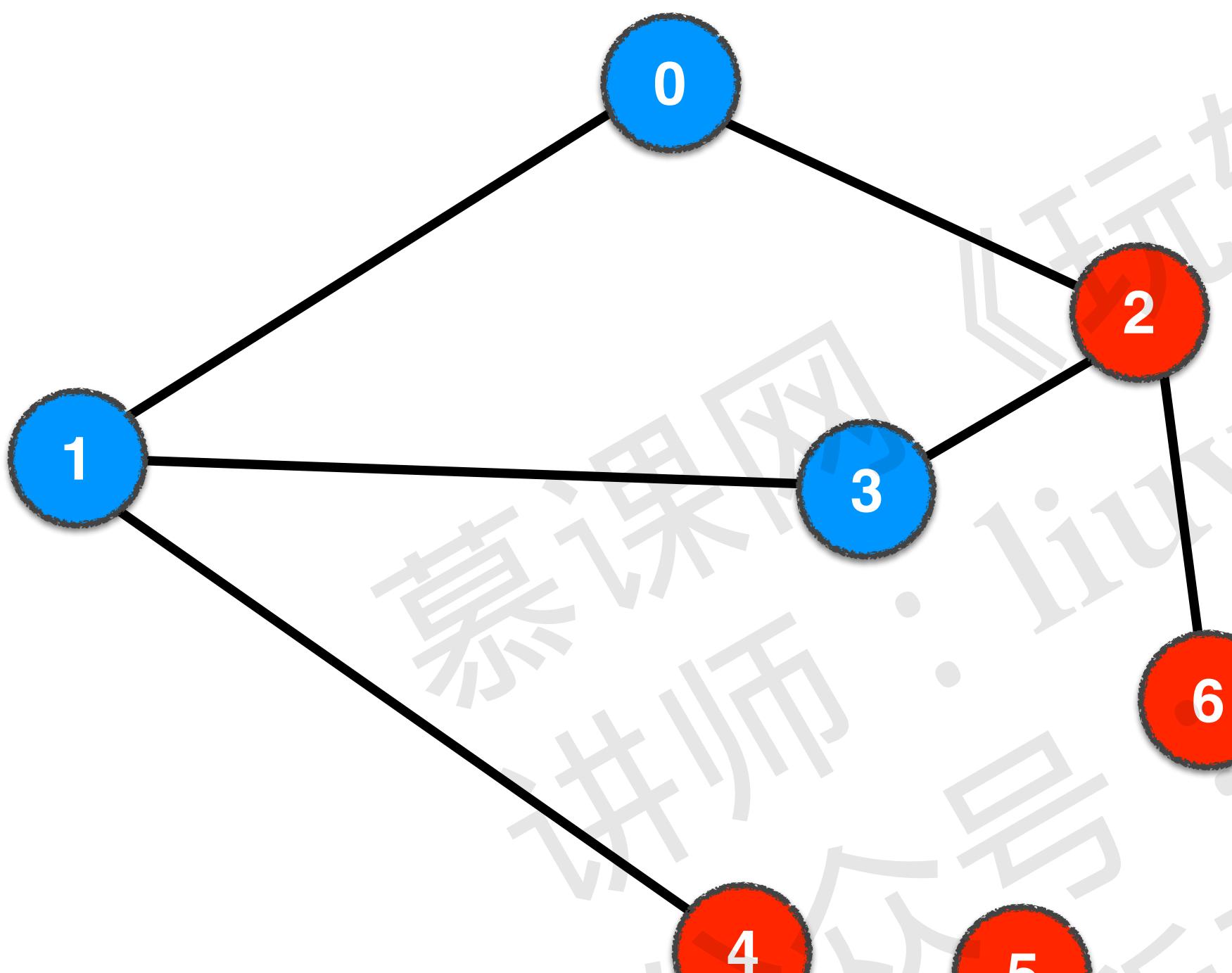
检测无向图中的环



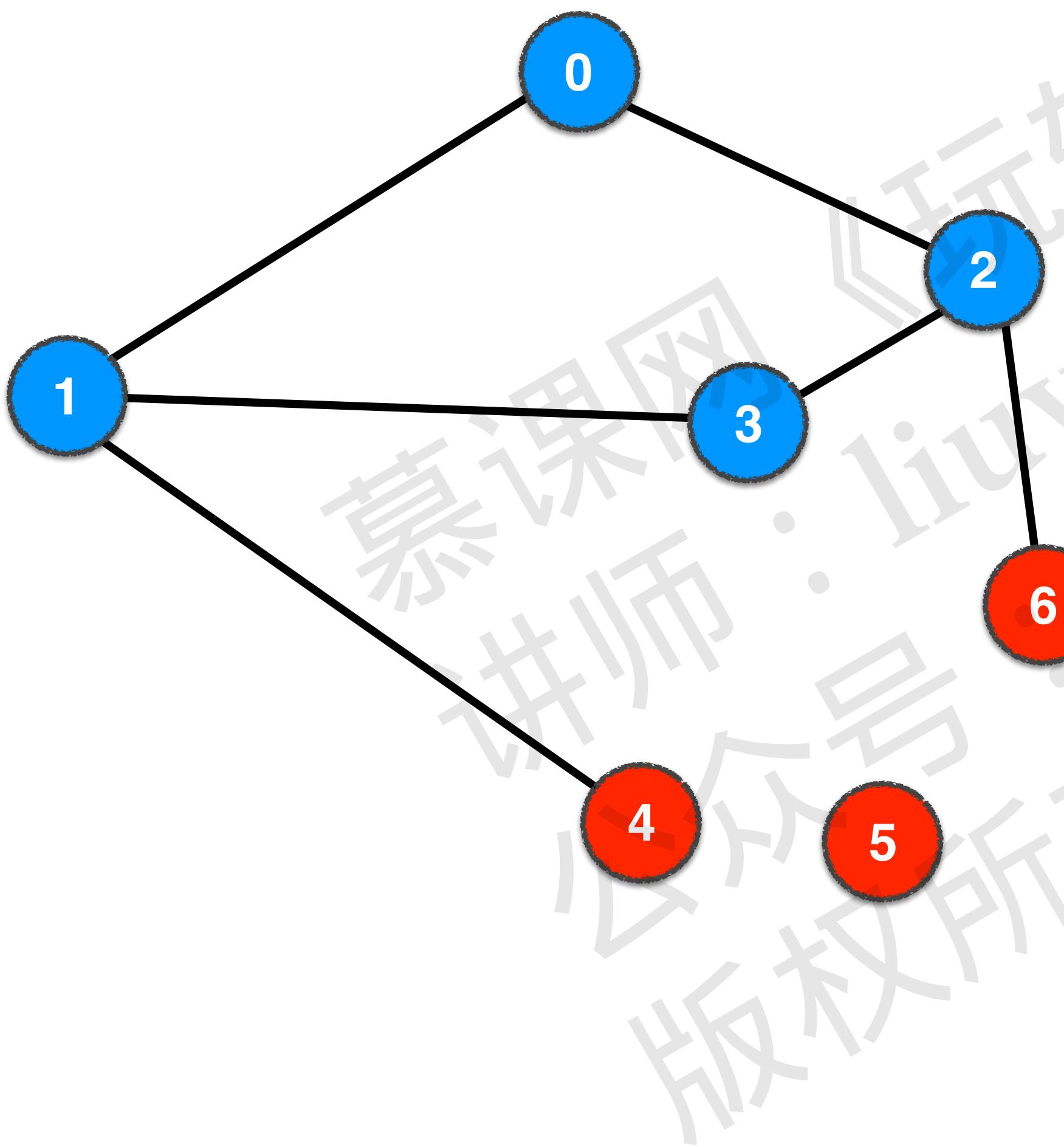
检测无向图中的环



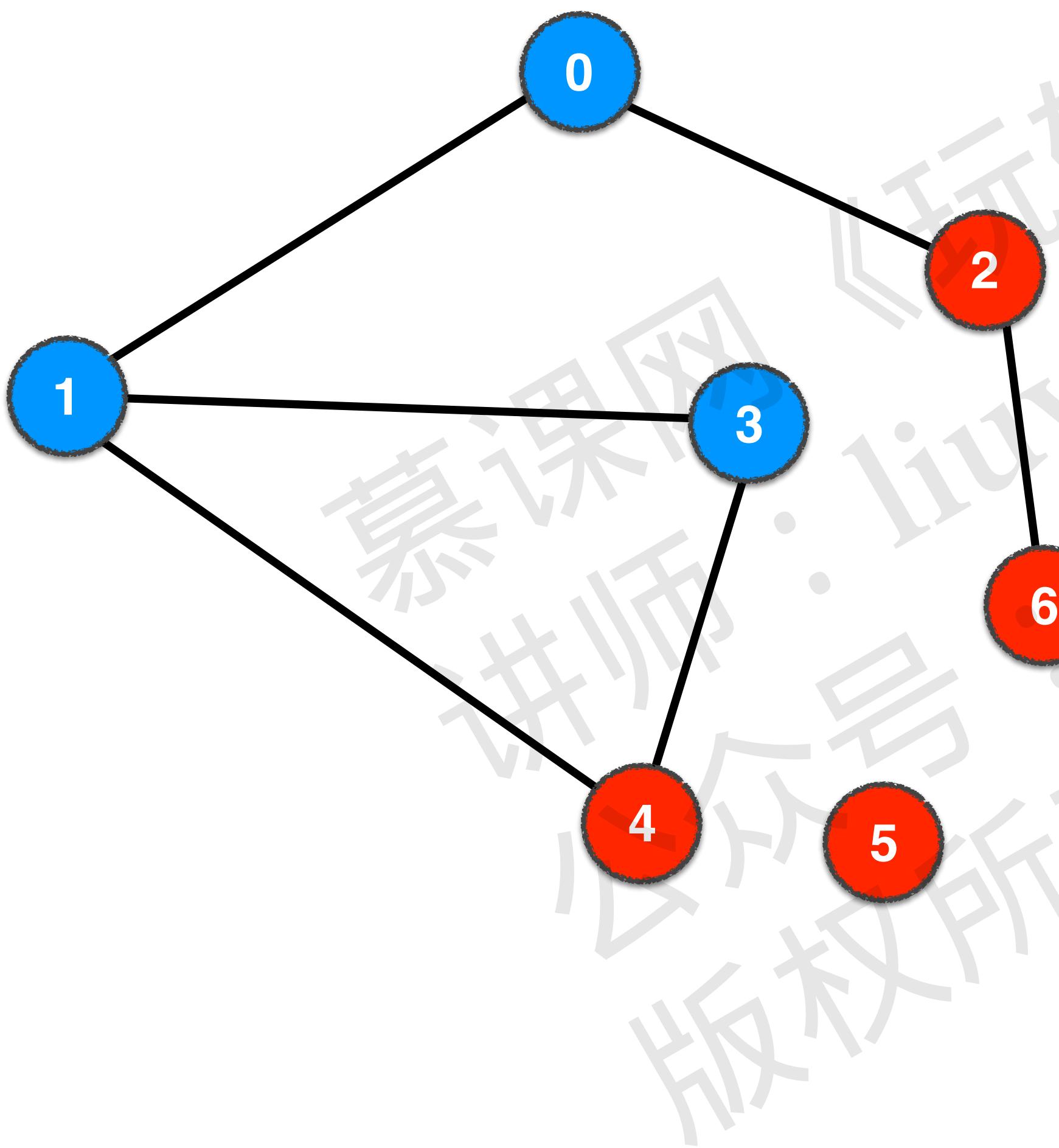
检测无向图中的环



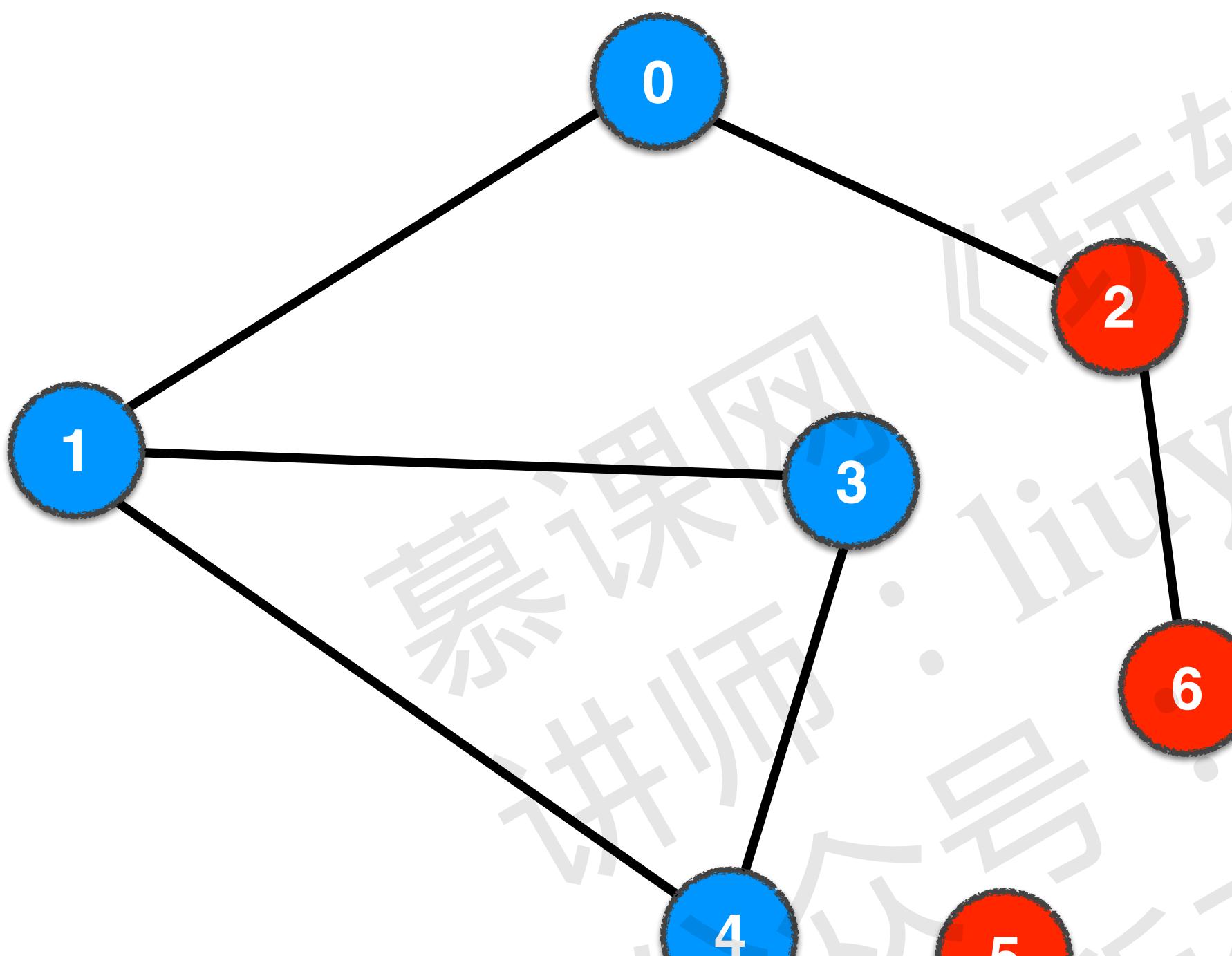
检测无向图中的环



检测无向图中的环



检测无向图中的环



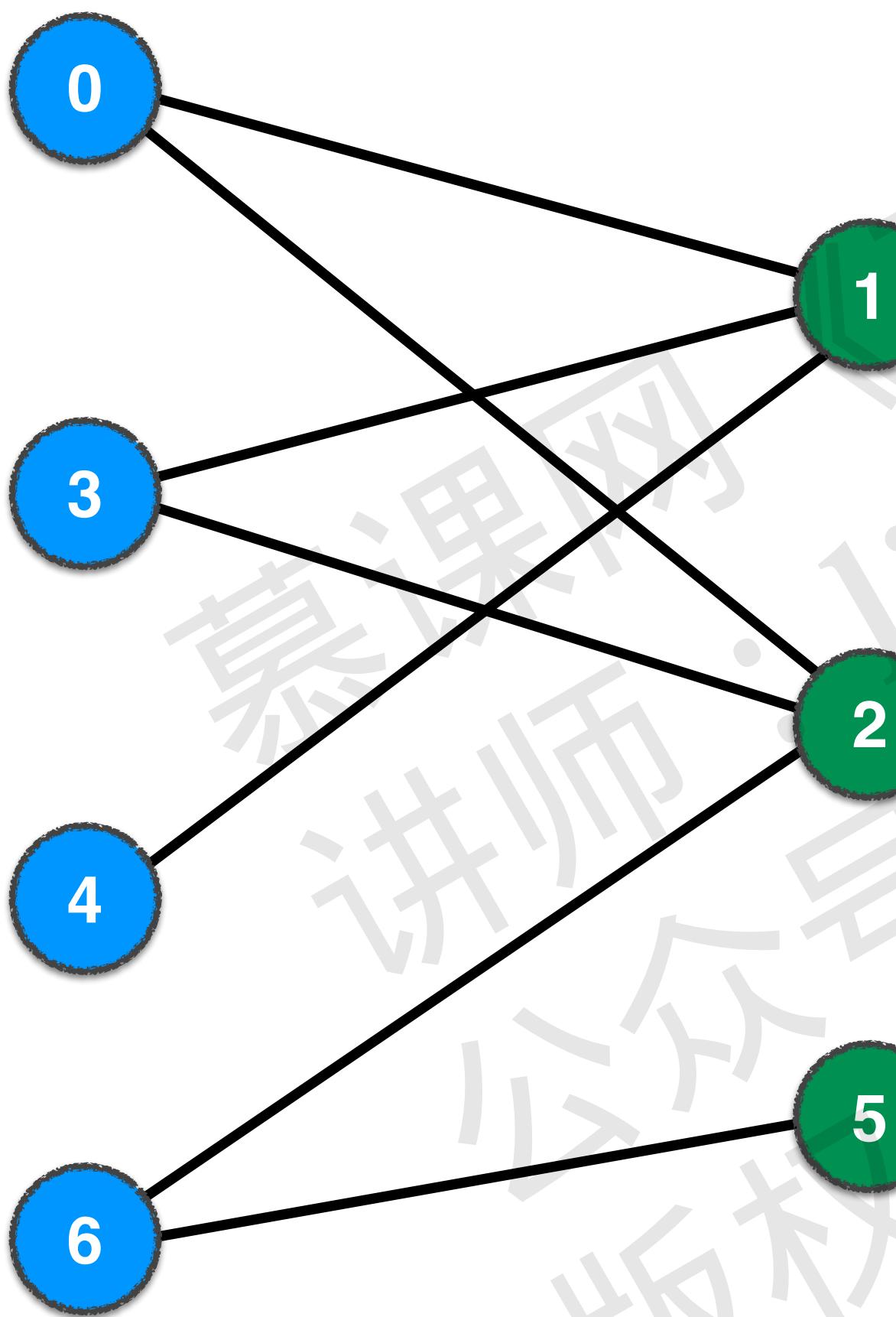
编程实践：无向图的环检测

慕课网 · liuyubobobo · 《玩转图论算法》

二分图检测

liuyubobobo

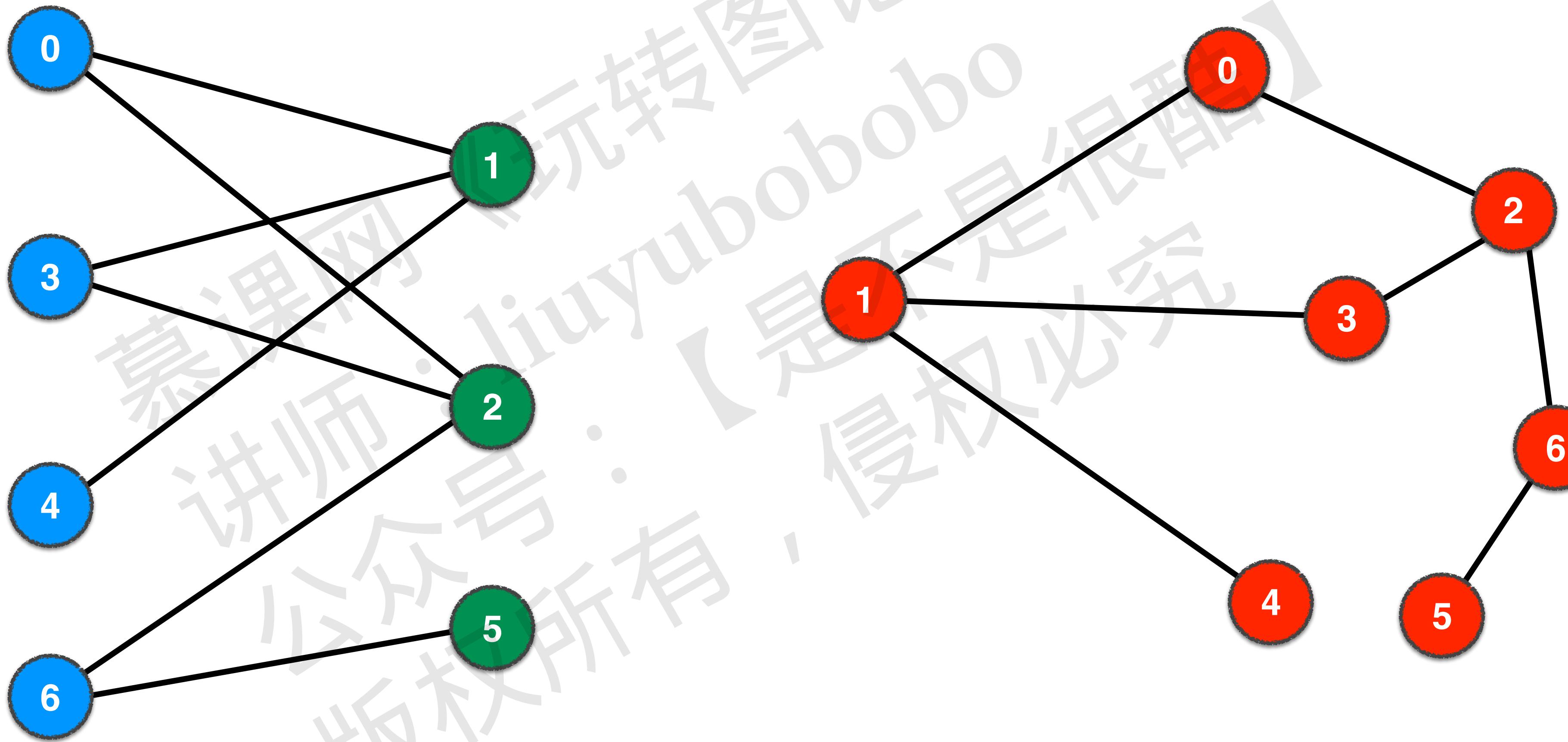
二分图检测



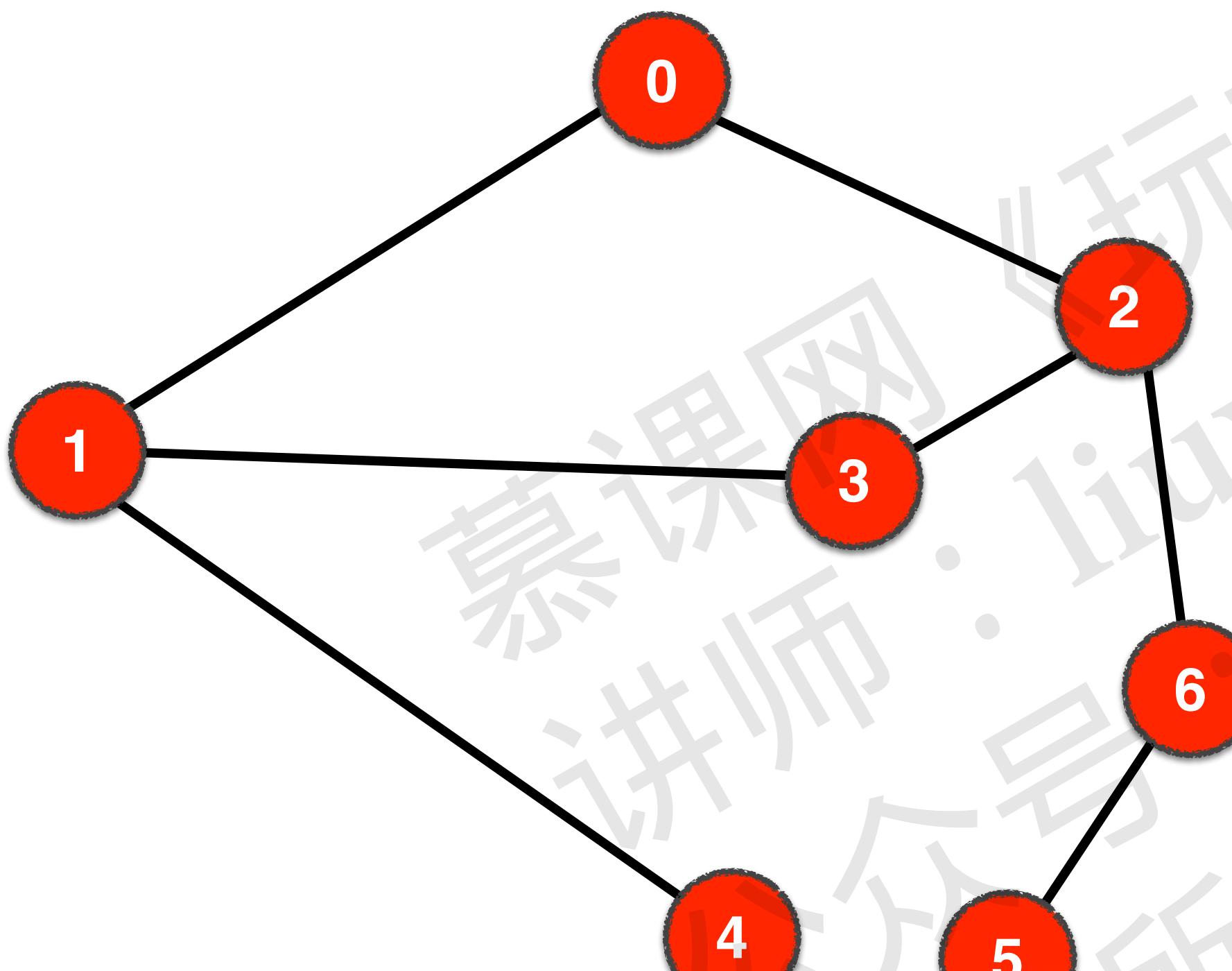
顶点 V 可以分成不相交的两部分

所有的边的两个端点隶属
不同的部分

二分图检测

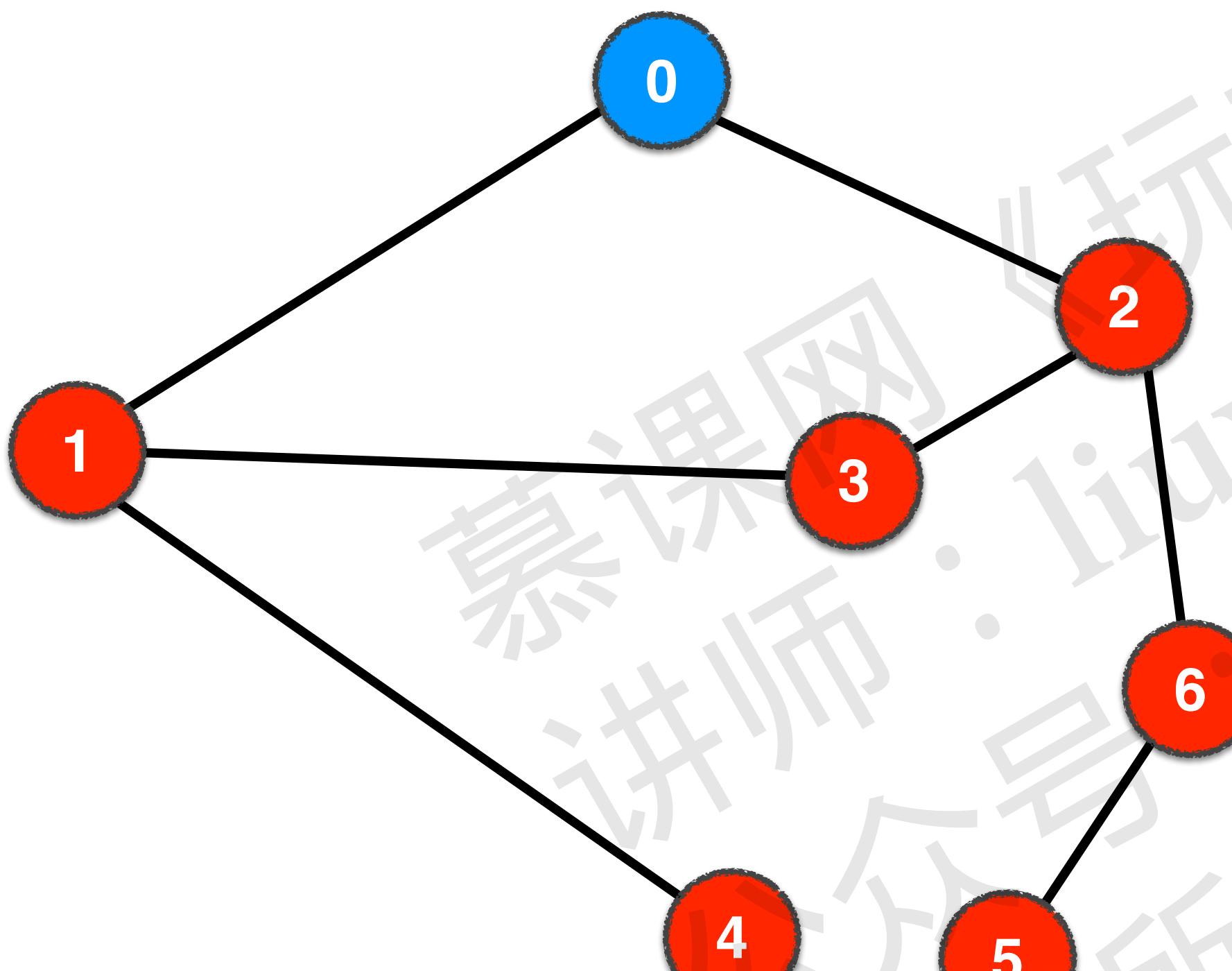


二分图检测



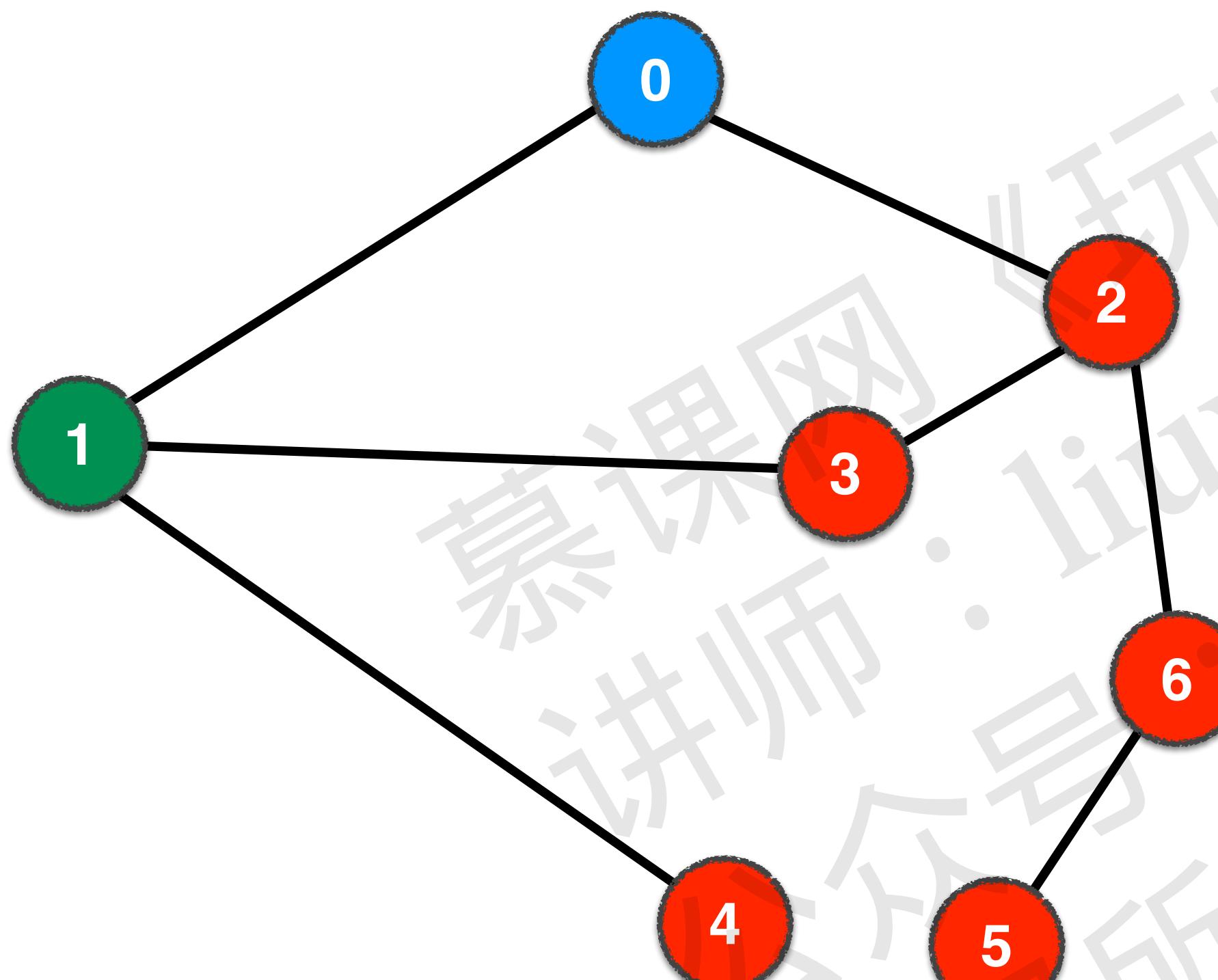
染色

二分图检测



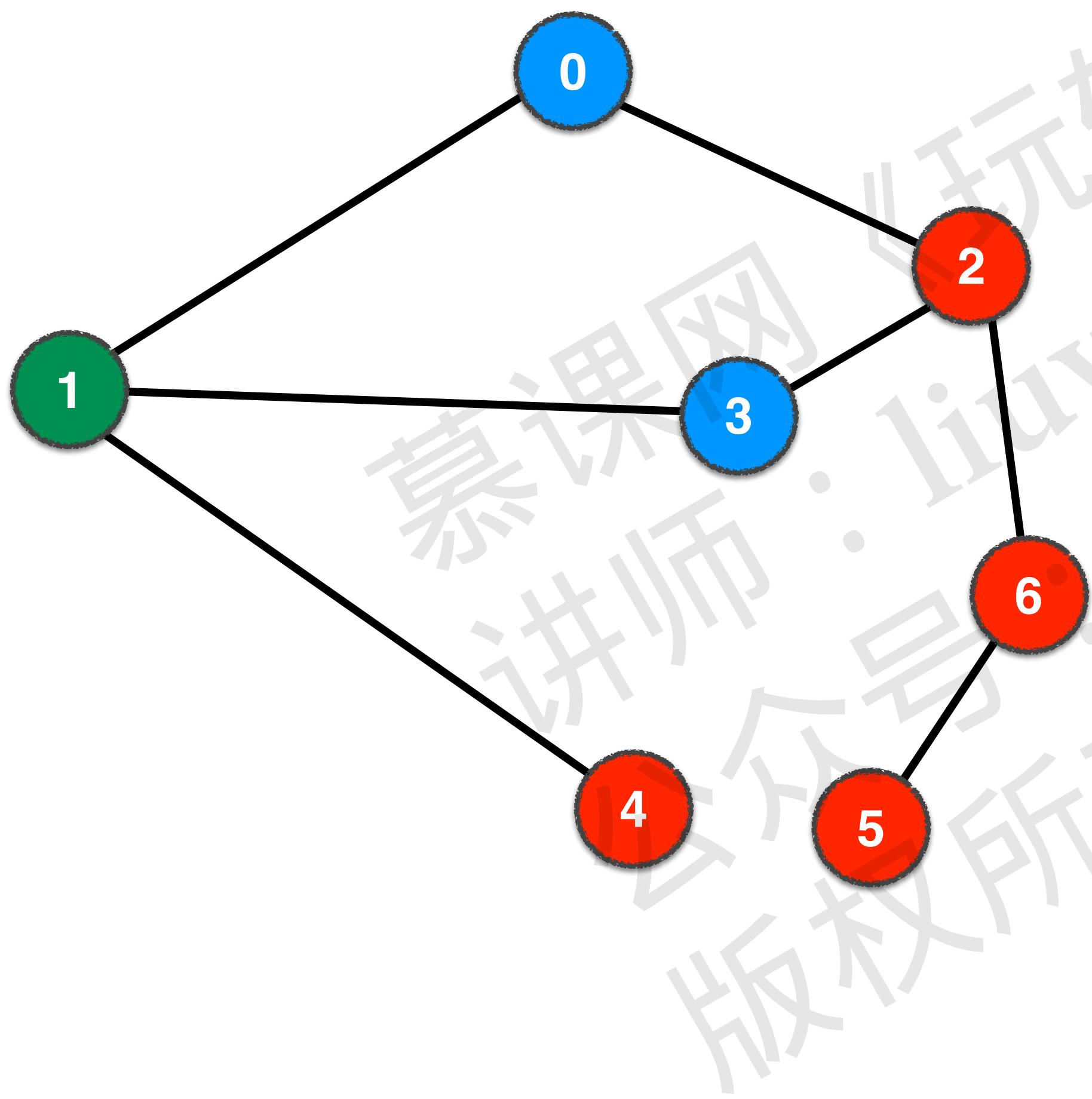
染色

二分图检测



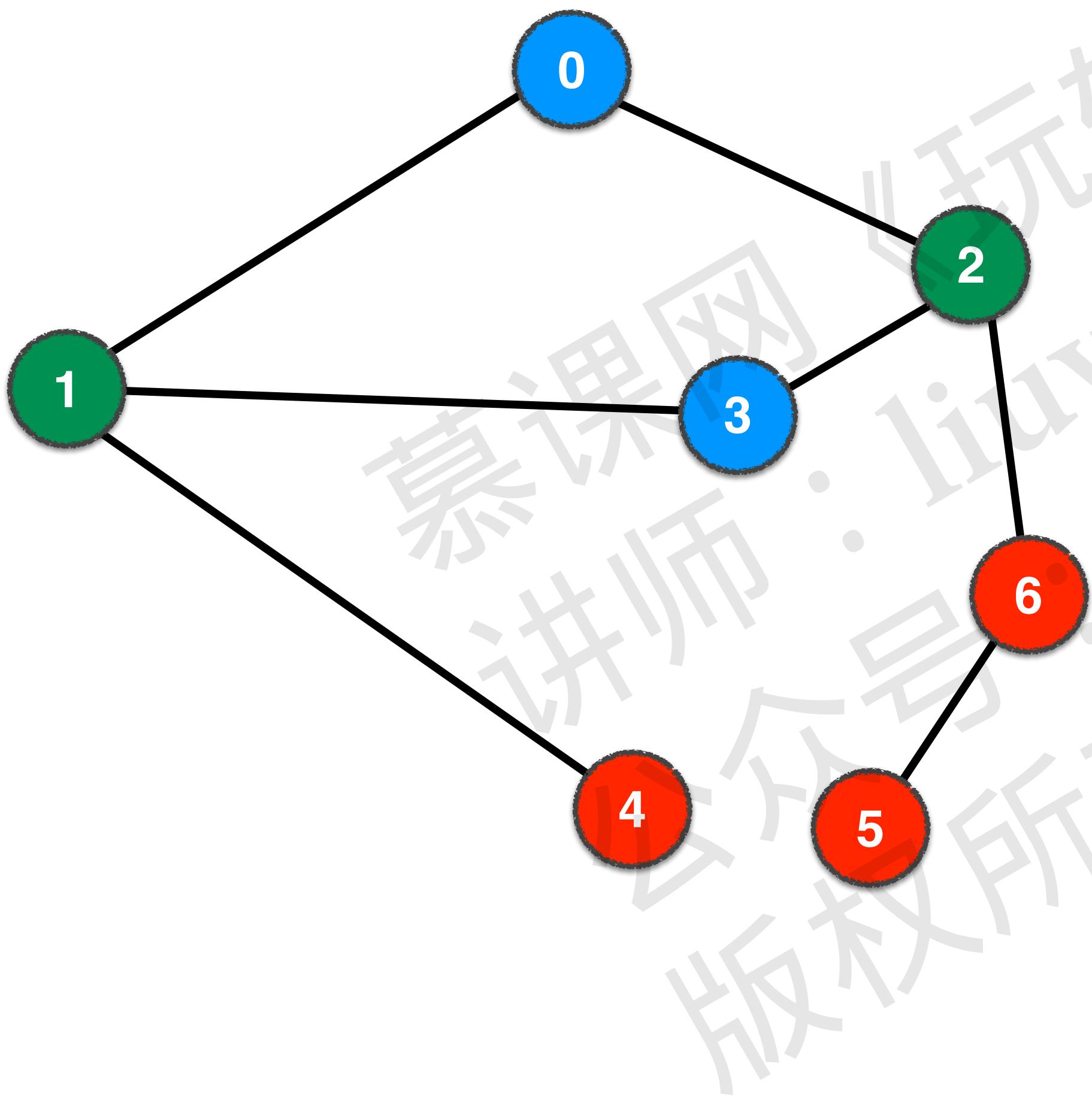
染色

二分图检测



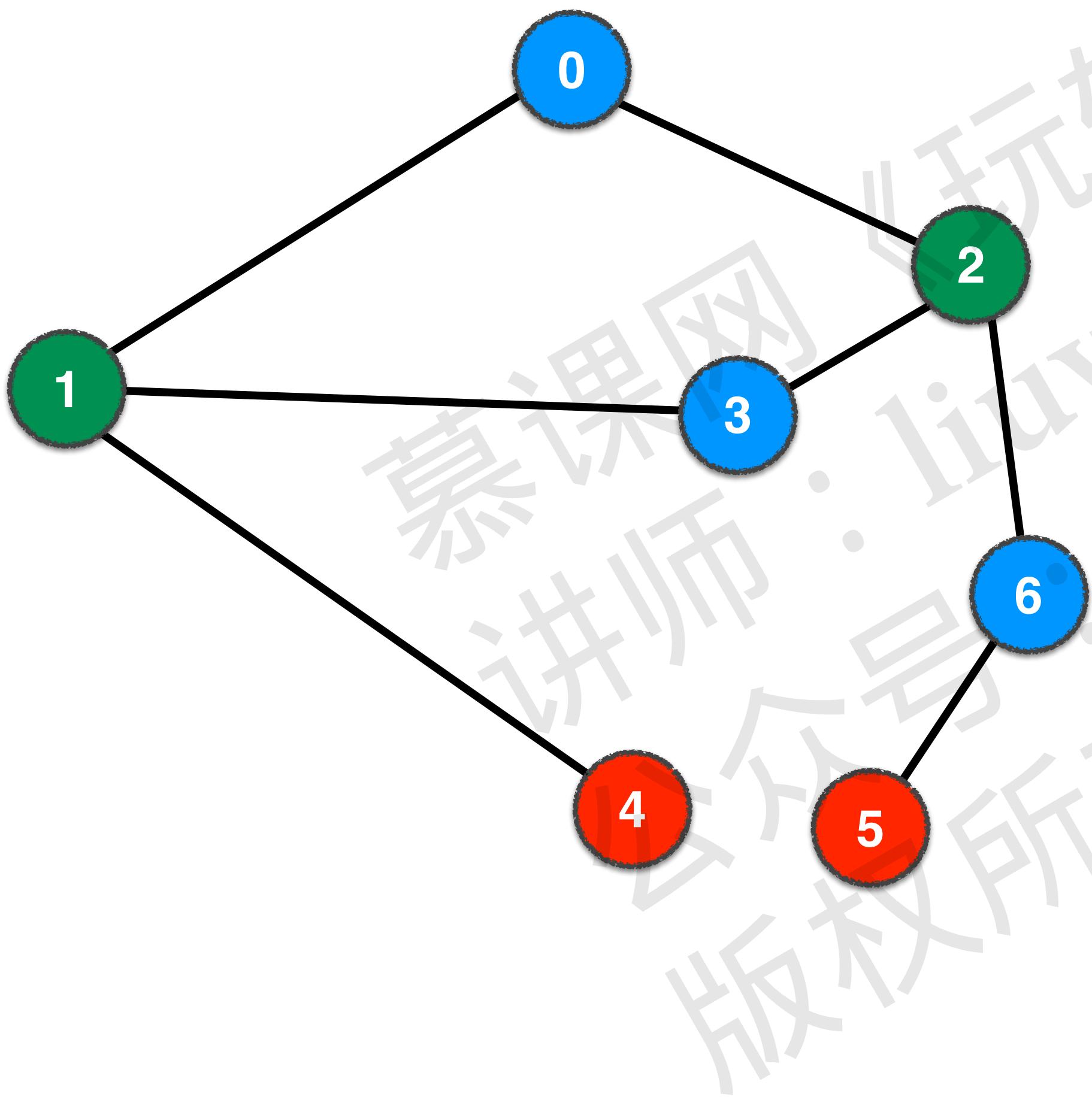
染色

二分图检测



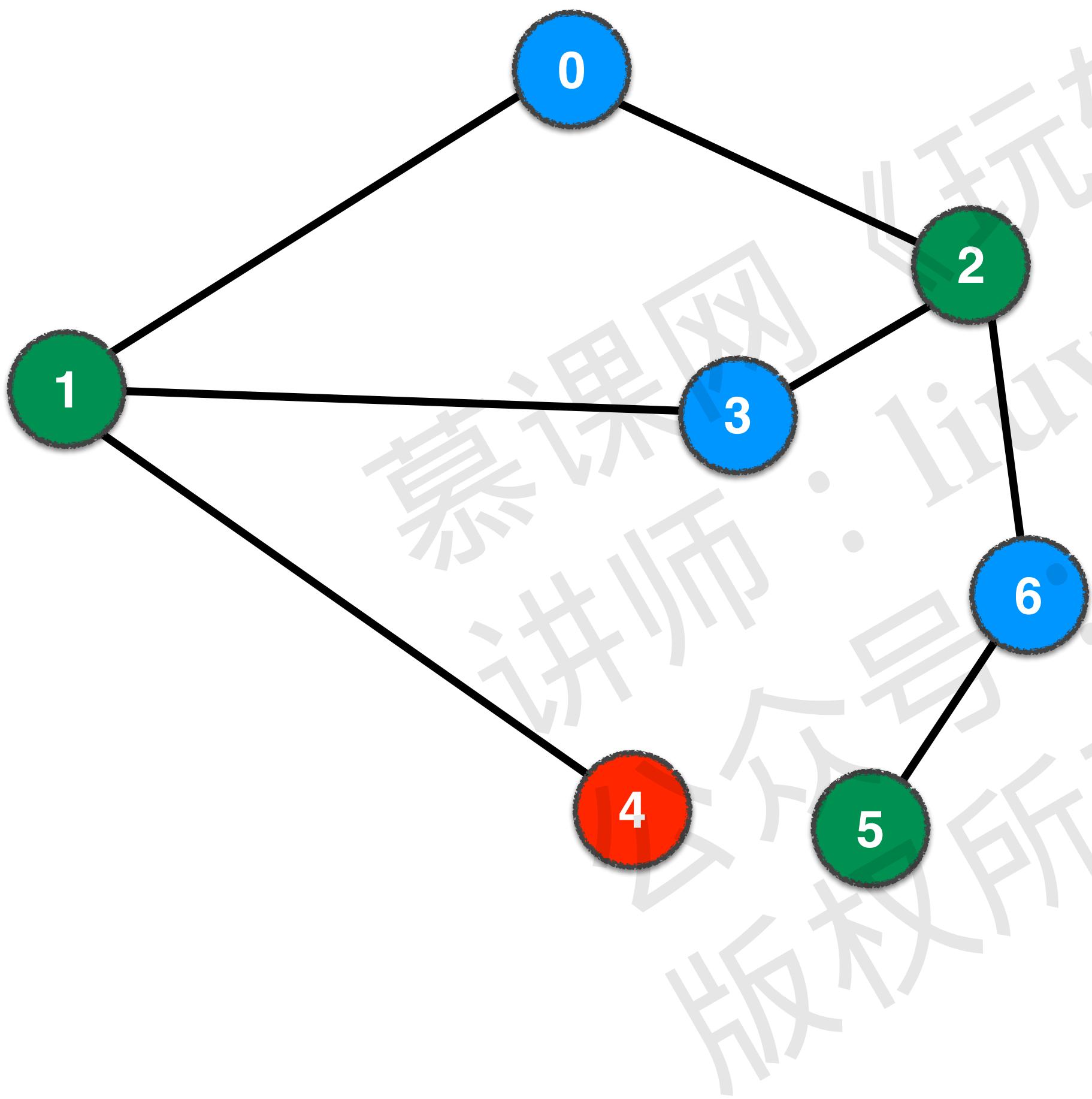
染色

二分图检测



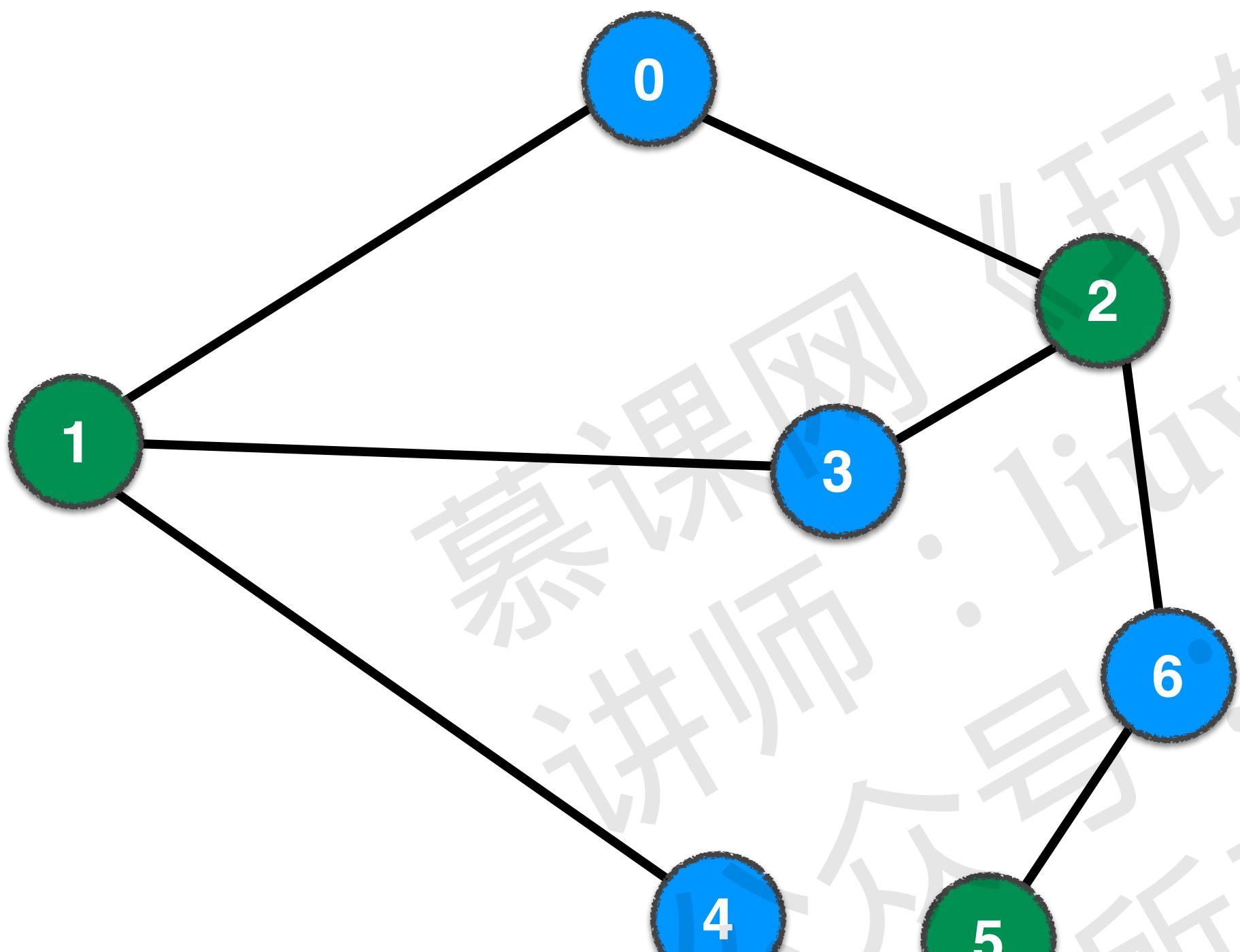
染色

二分图检测



染色

二分图检测



染色

二分图检测的实现

liuyubobobo

编程实践：二分图检测

慕课网 · 讲师：liuyubobobo · 《玩转图论算法》

本章小结和更多拓展

liuyubobobo

图的深度优先遍历

联通分量

路径问题（单源路径）

环检测

二分图检测

图的深度优先遍历

记录更多信息

联通分量

ccid

路径问题（单源路径）

pre

环检测

colors

二分图检测

图的深度优先遍历

递归返回值

联通分量

路径问题（单源路径）

环检测

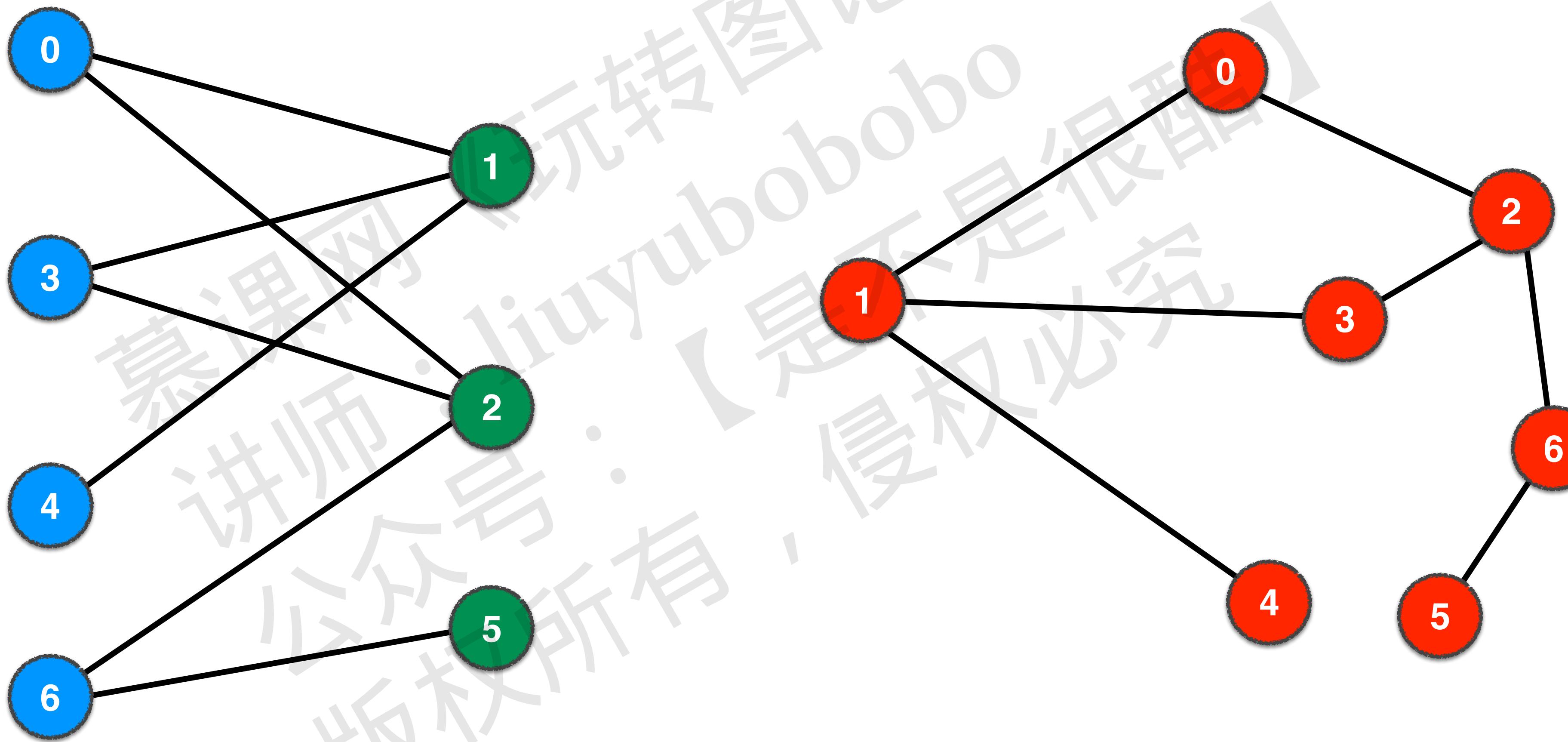
二分图检测



图的深度优先遍历

更多图的深度优先遍历的应用，后续课程还会涉及

更多拓展

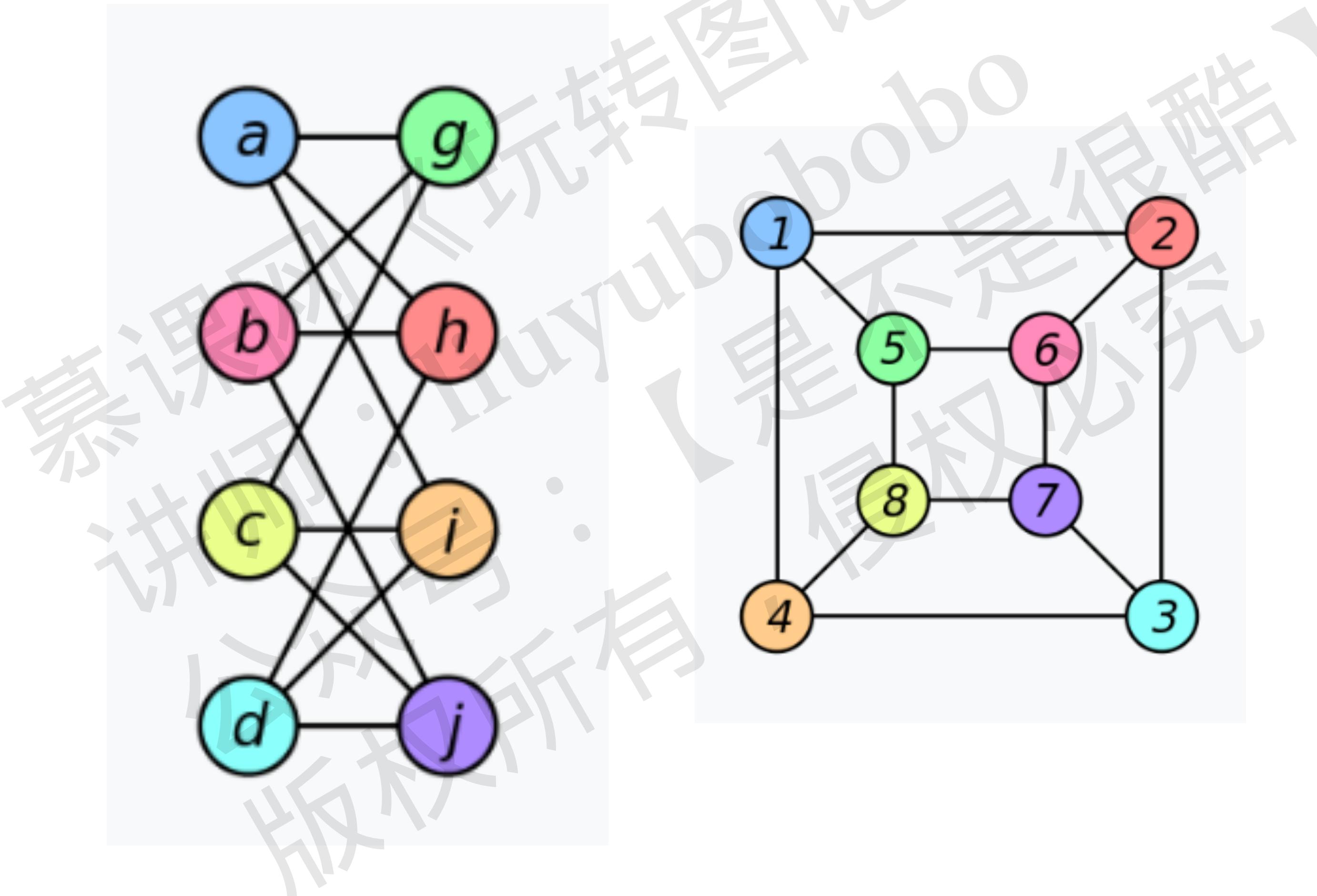


图同构

更多拓展

图同构

NP难



$$\sigma(a) = 1$$

$$\sigma(b) = 6$$

$$\sigma(c) = 8$$

$$\sigma(d) = 3$$

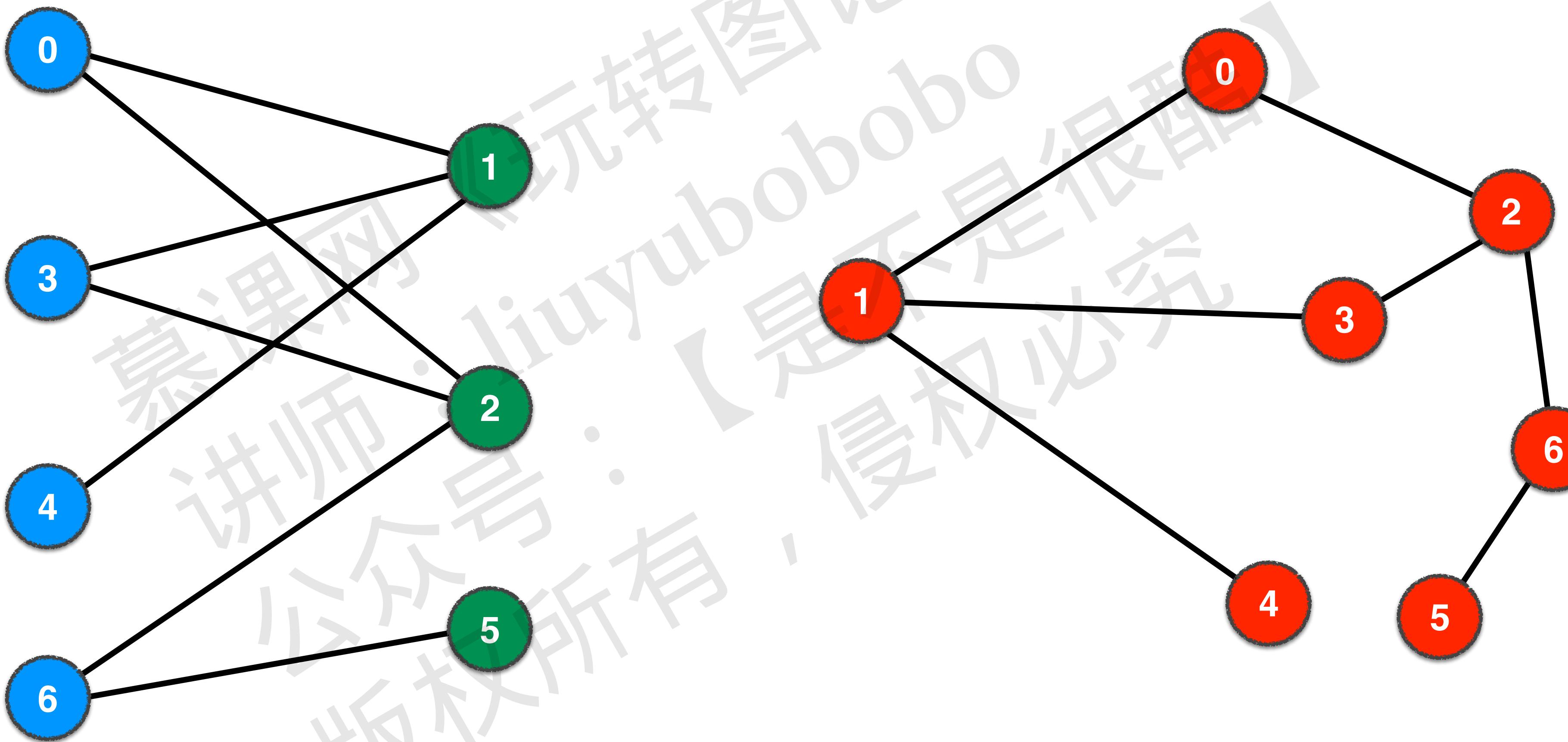
$$\sigma(g) = 5$$

$$\sigma(h) = 2$$

$$\sigma(i) = 4$$

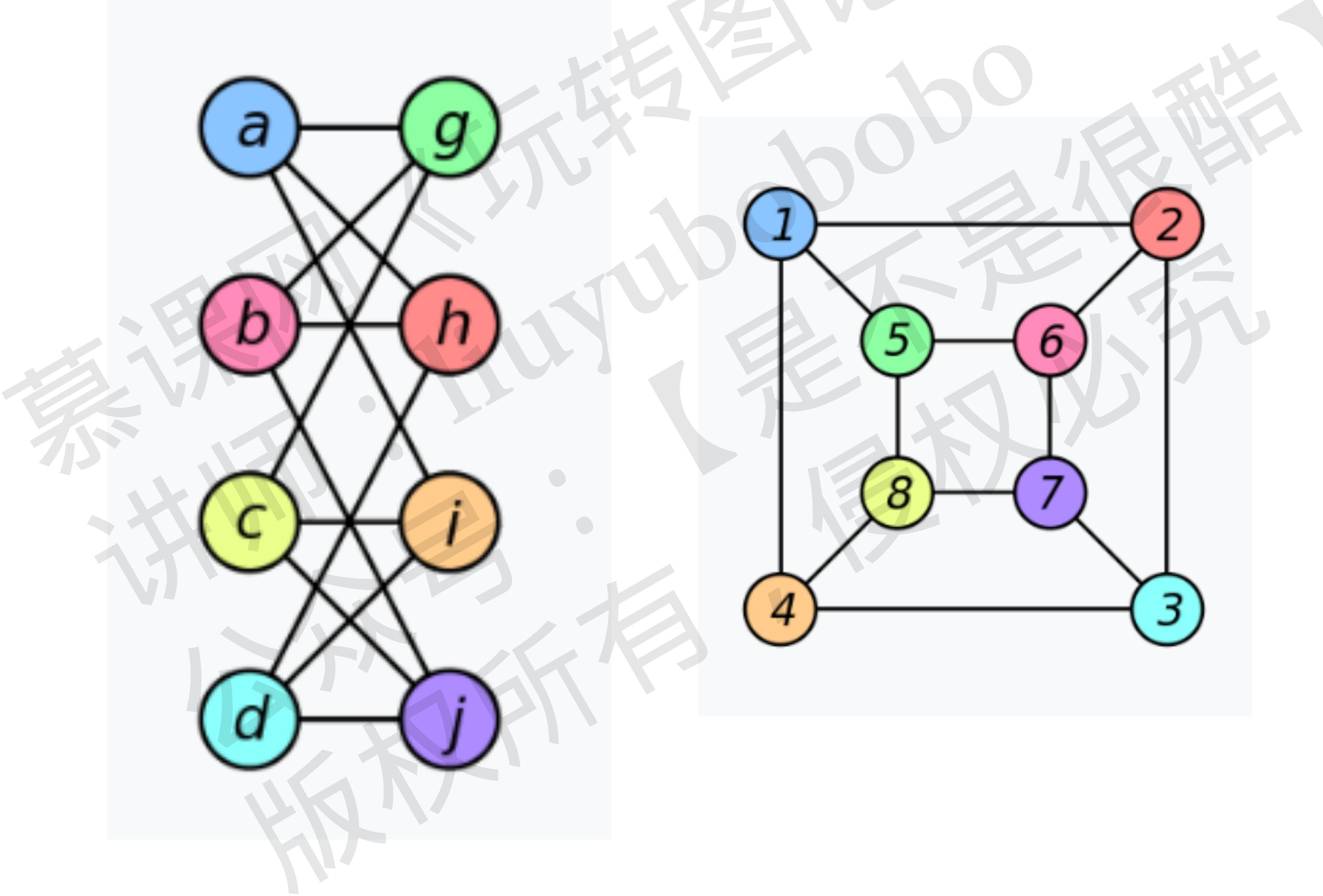
$$\sigma(j) = 7$$

更多拓展

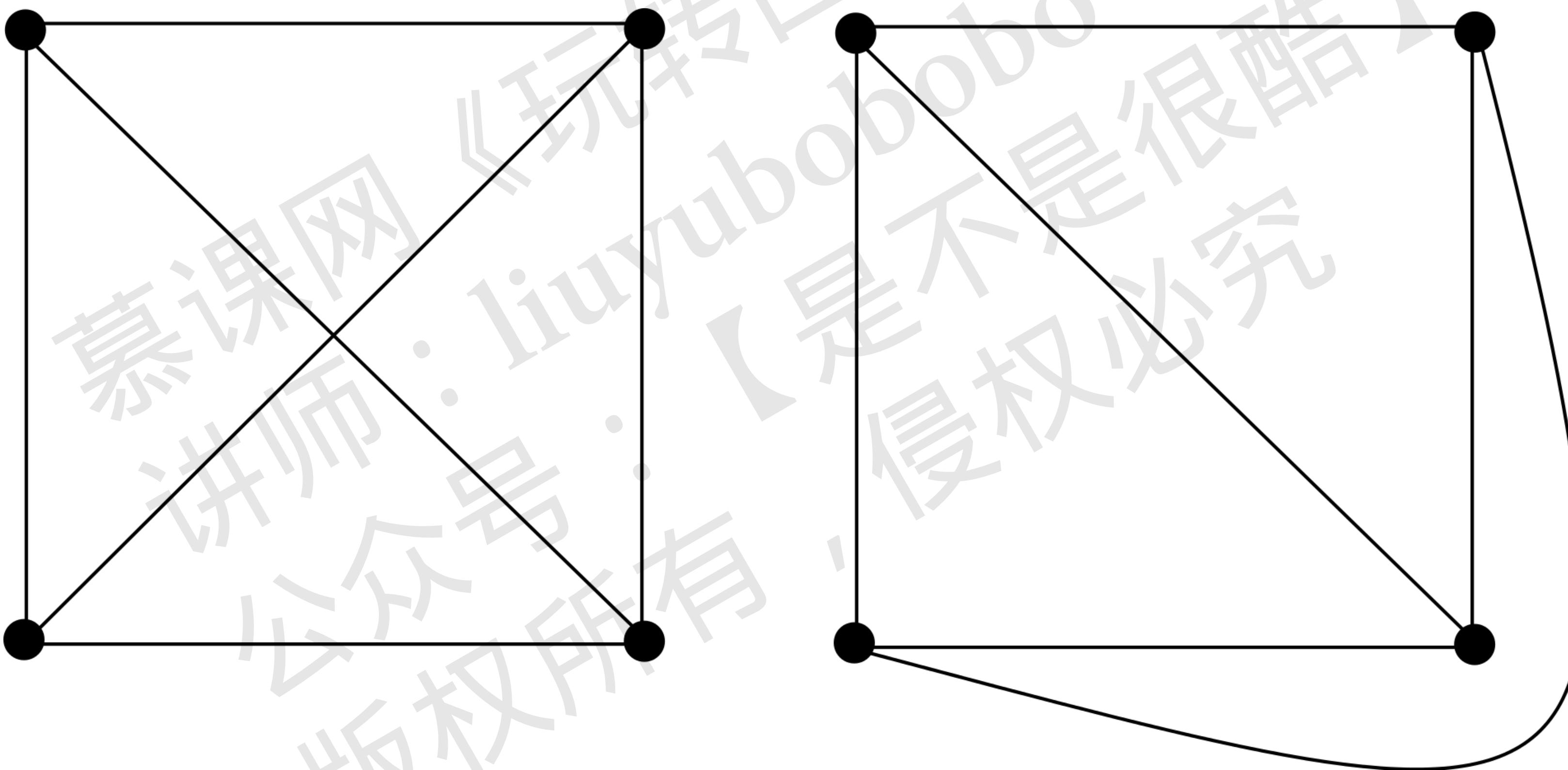


平面图

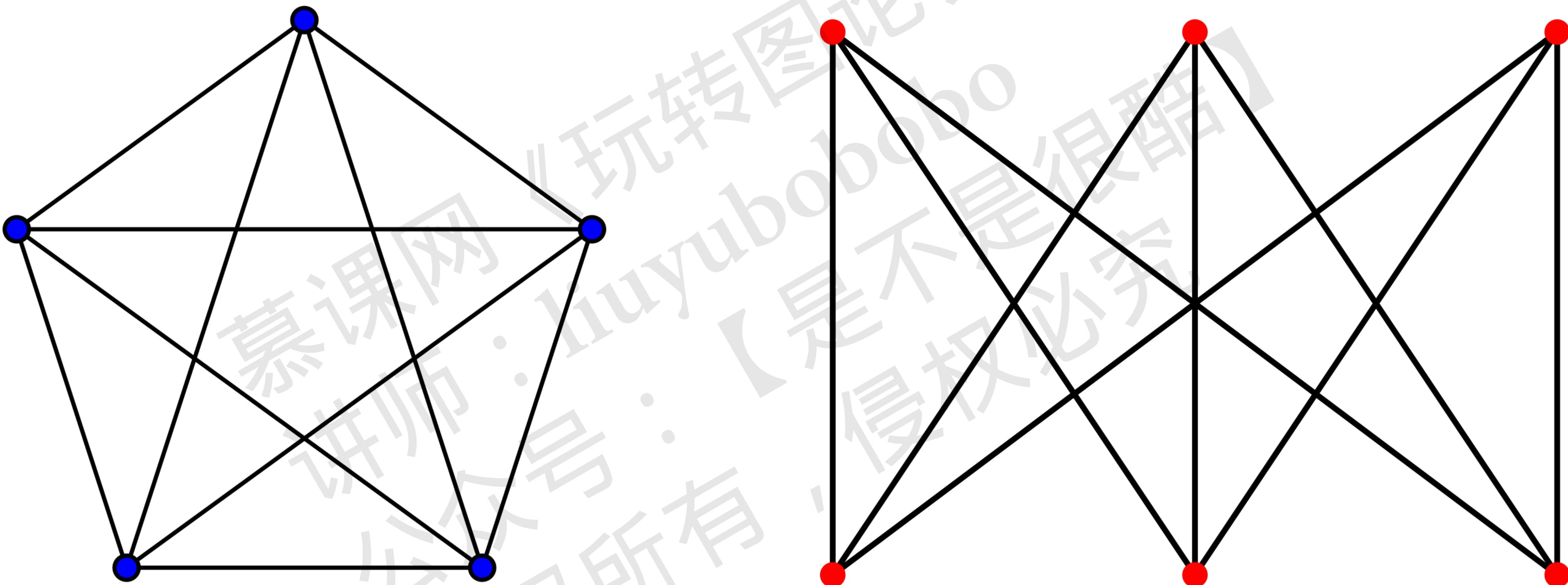
更多拓展



更多拓展



更多拓展



偏数学理论研究，这个课程以算法实现为主：)

大家加油！

欢迎大家关注我的个人公众号：是不是很酷



「是不是很酷」

坚持有质量的技术原创

用技术人的视角看世界

玩儿转图论算法

liuyubobobo