

玩儿转图论算法

liuyubobobo

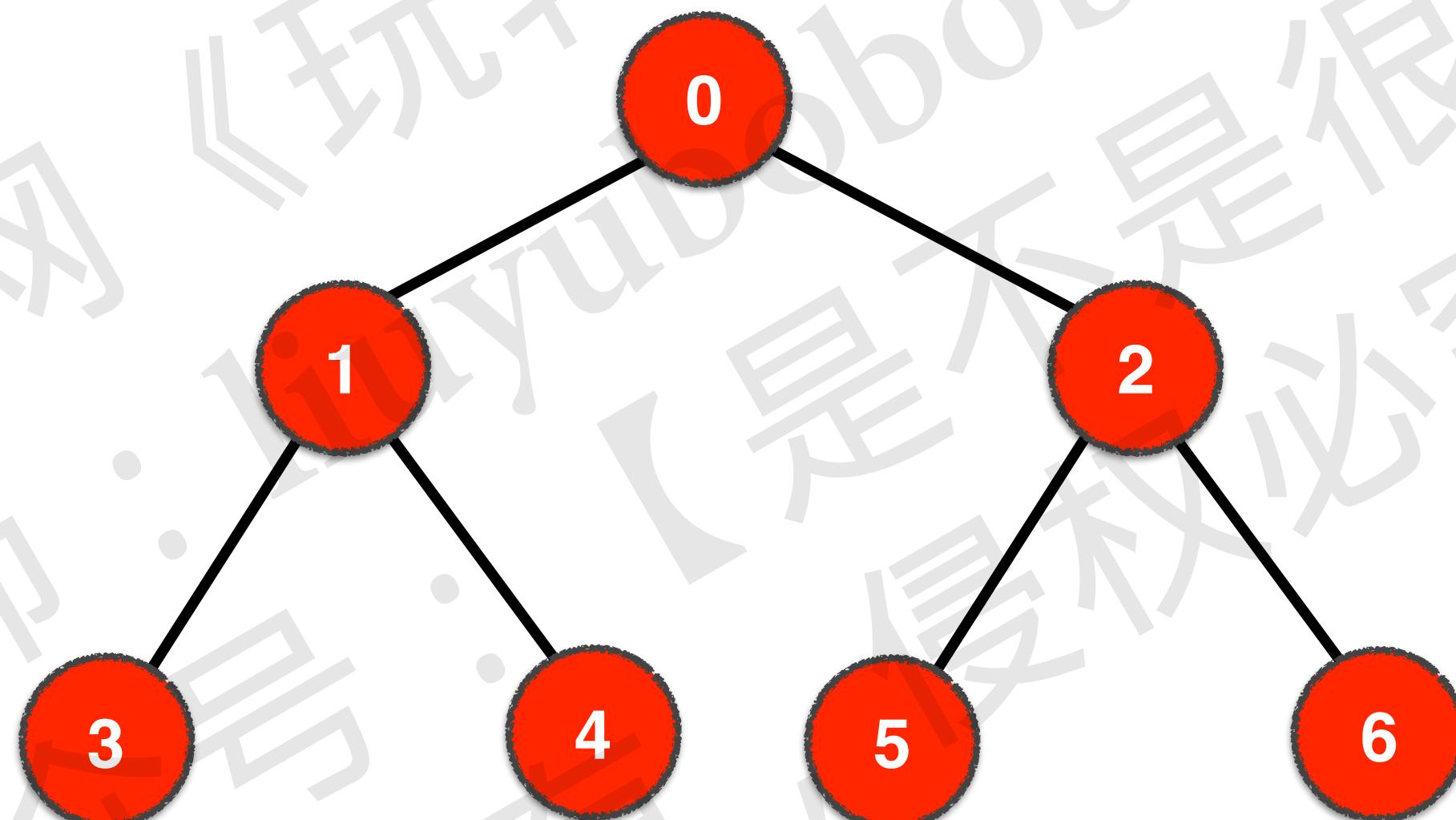
图的广度优先遍历

liuyubobobo

从树的广度优先遍历 到图的广度优先遍历

liuyubobobo

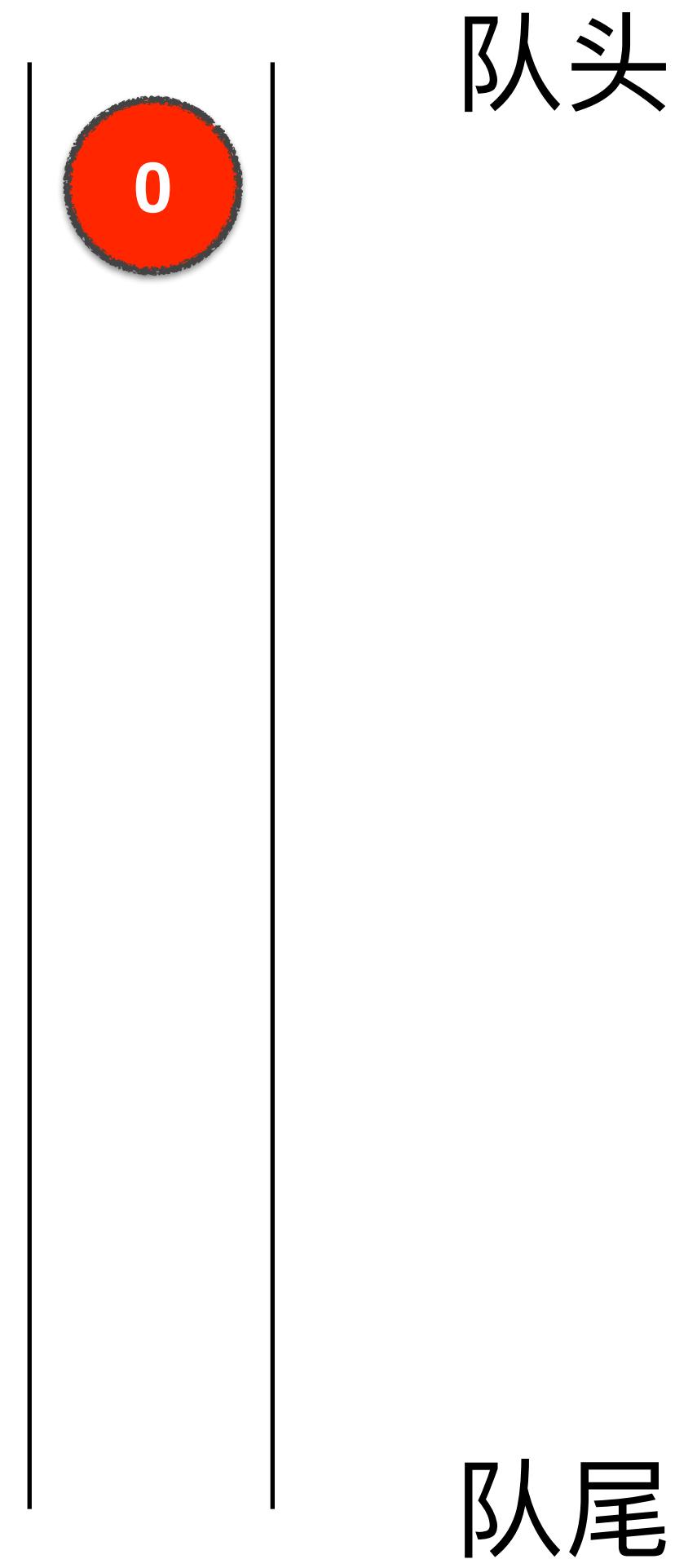
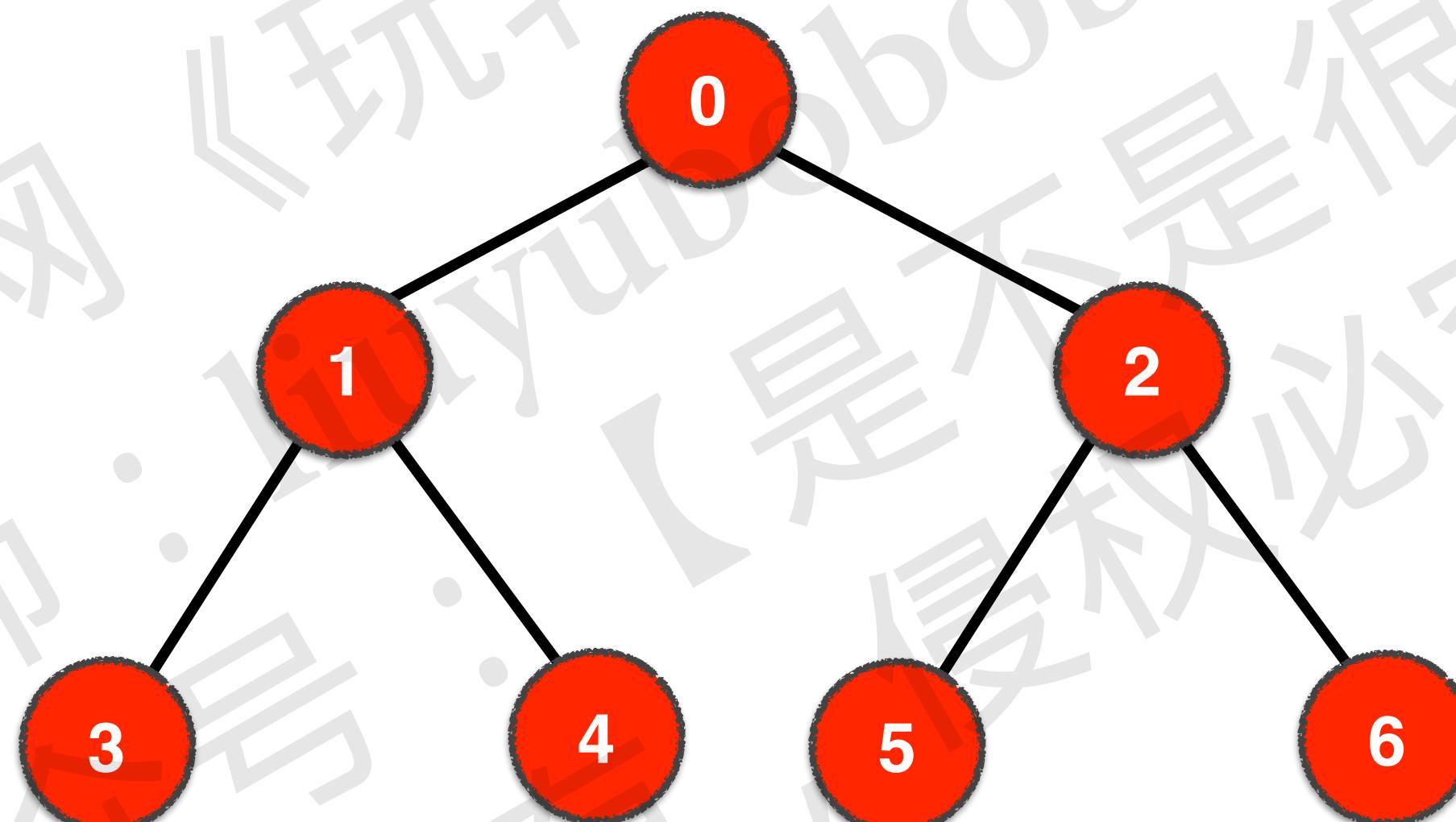
回顾树的广度优先遍历



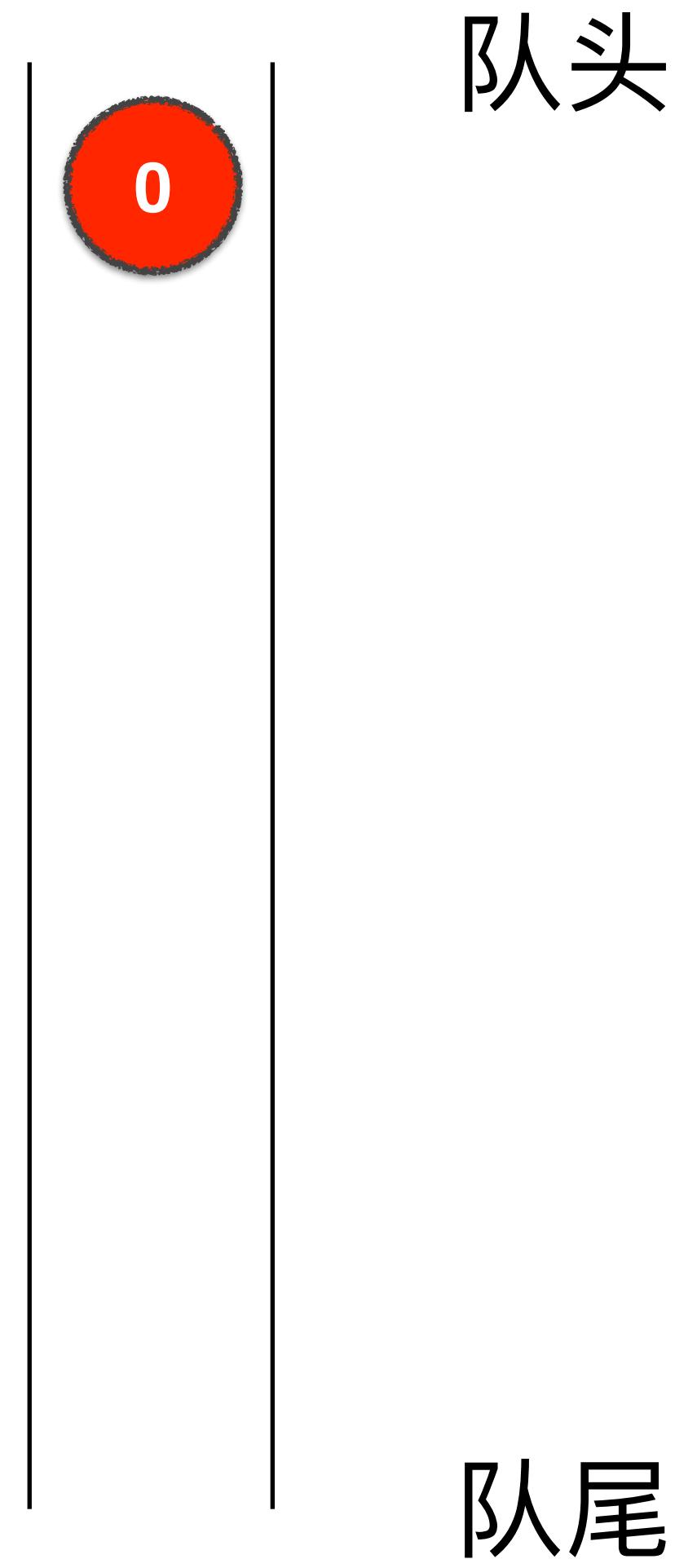
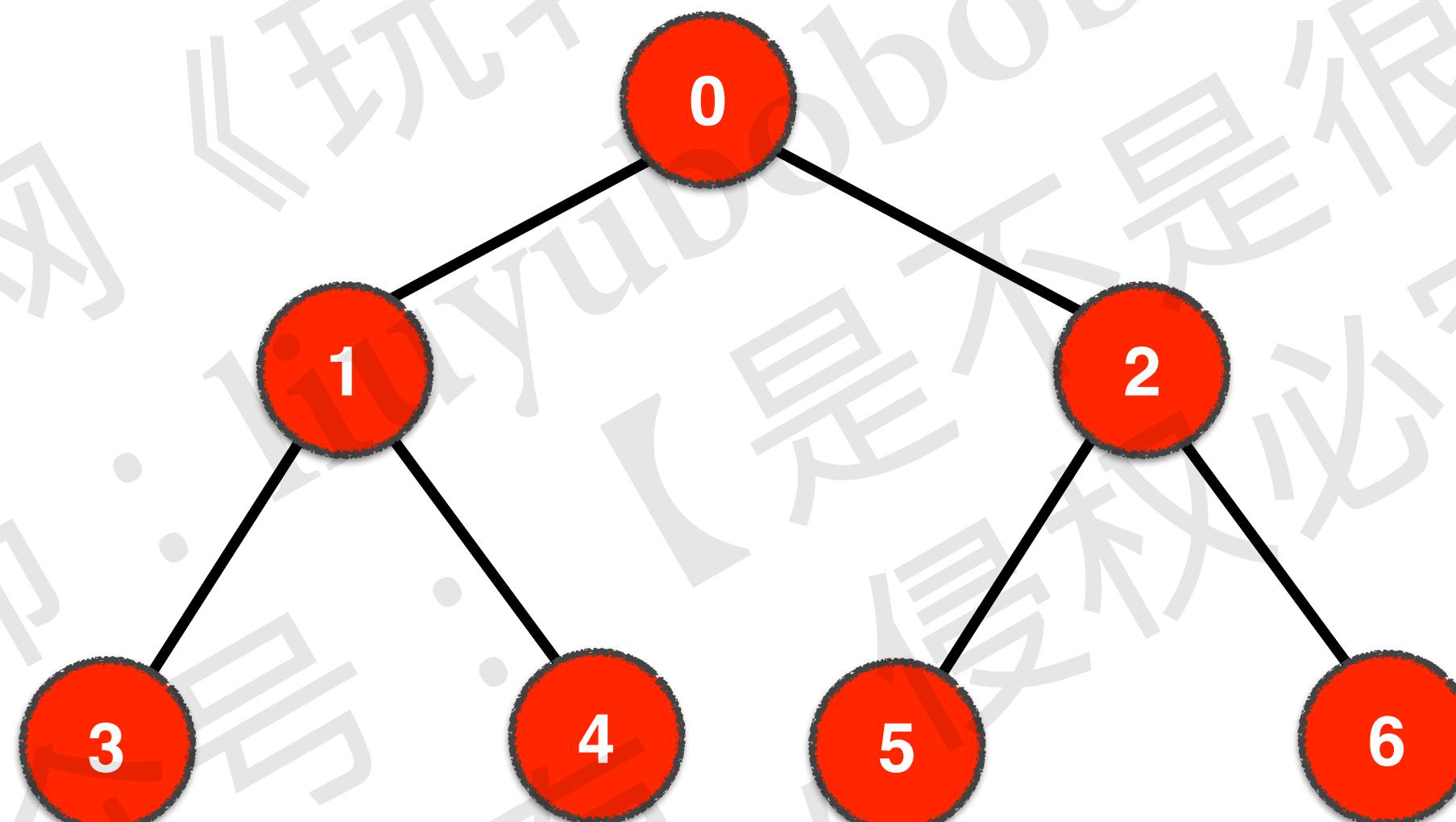
队头

队尾

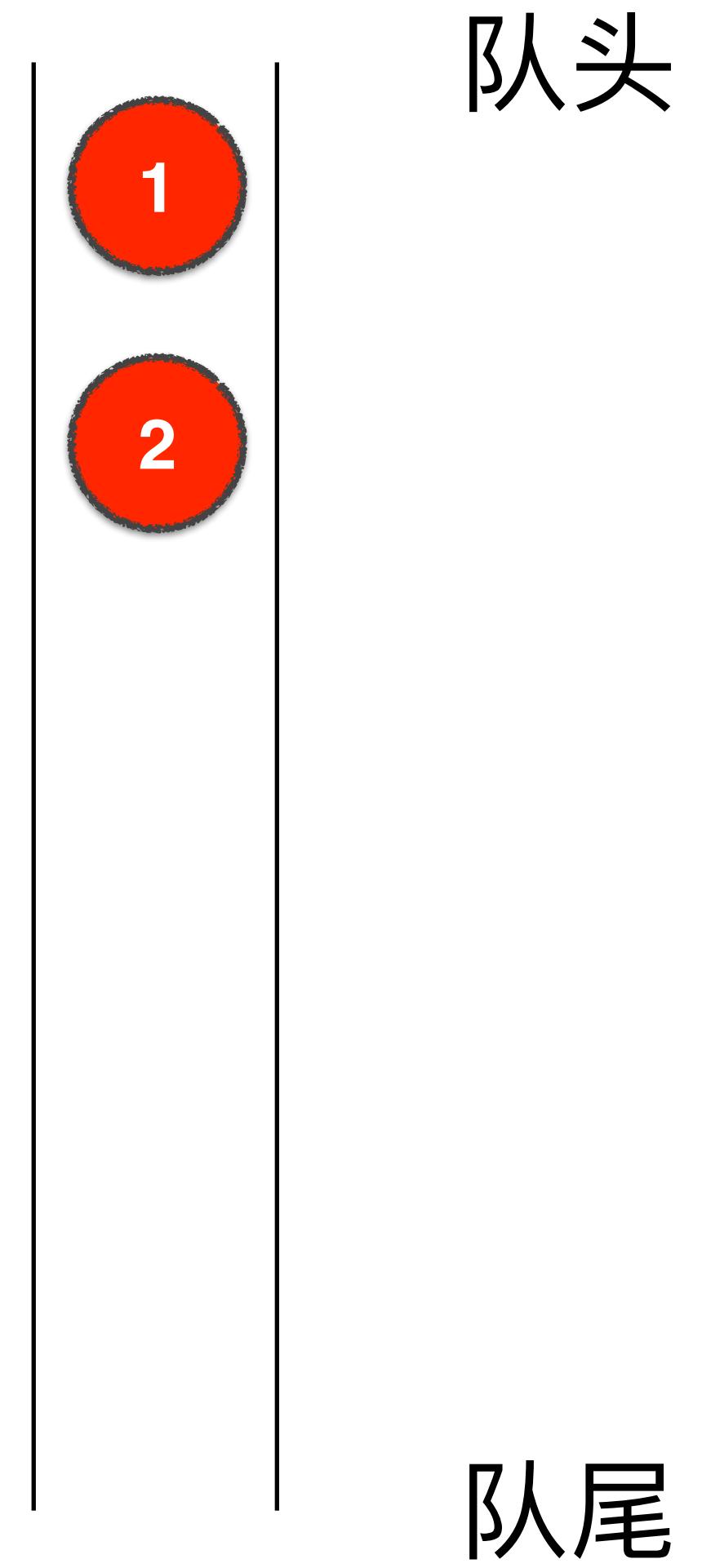
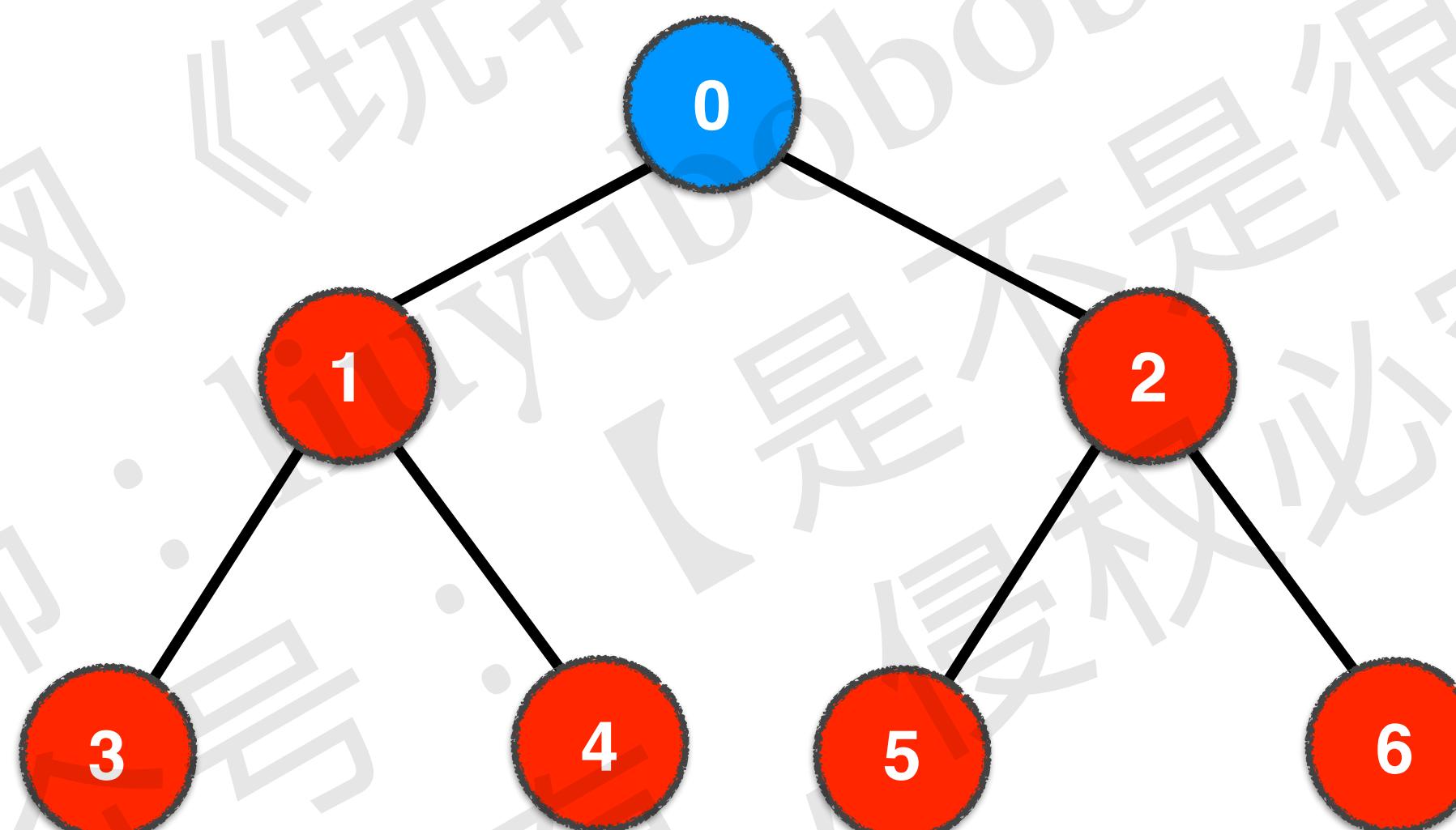
回顾树的广度优先遍历



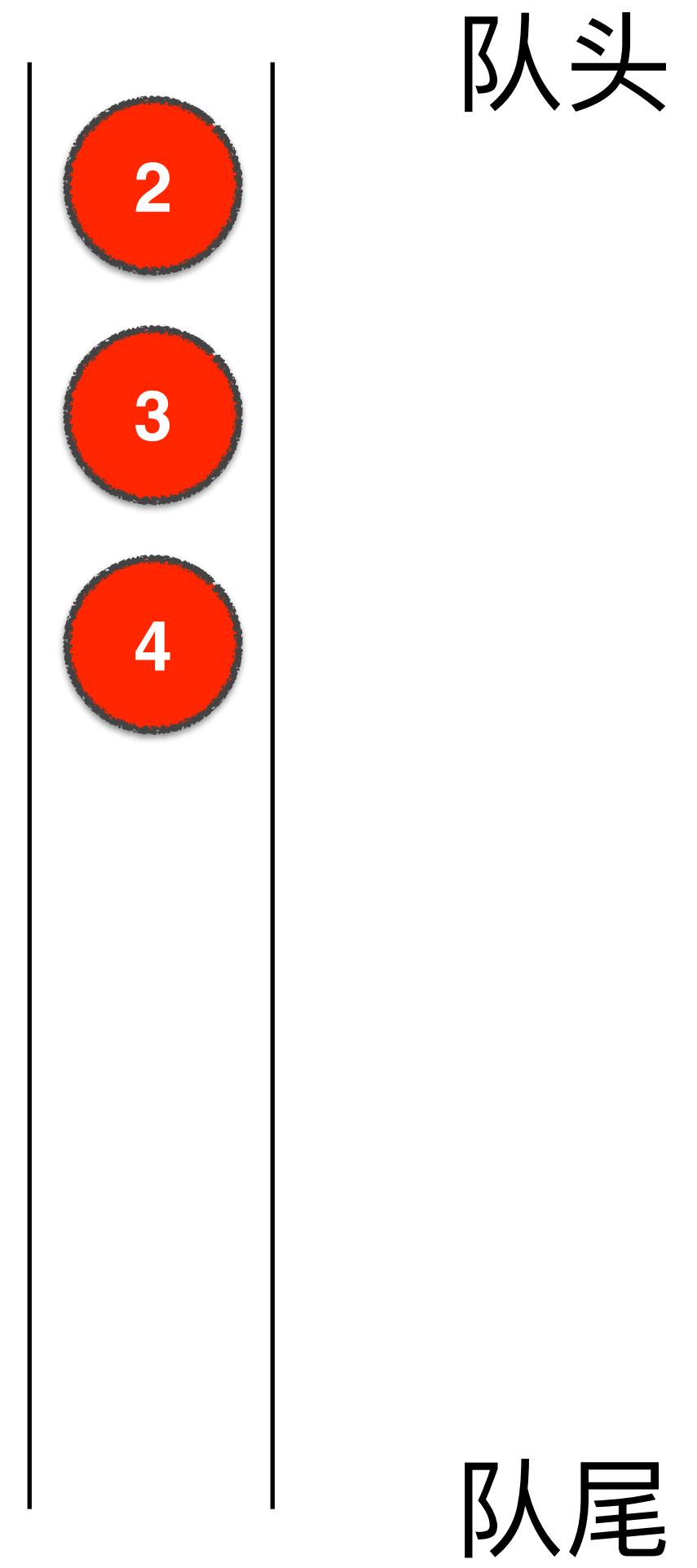
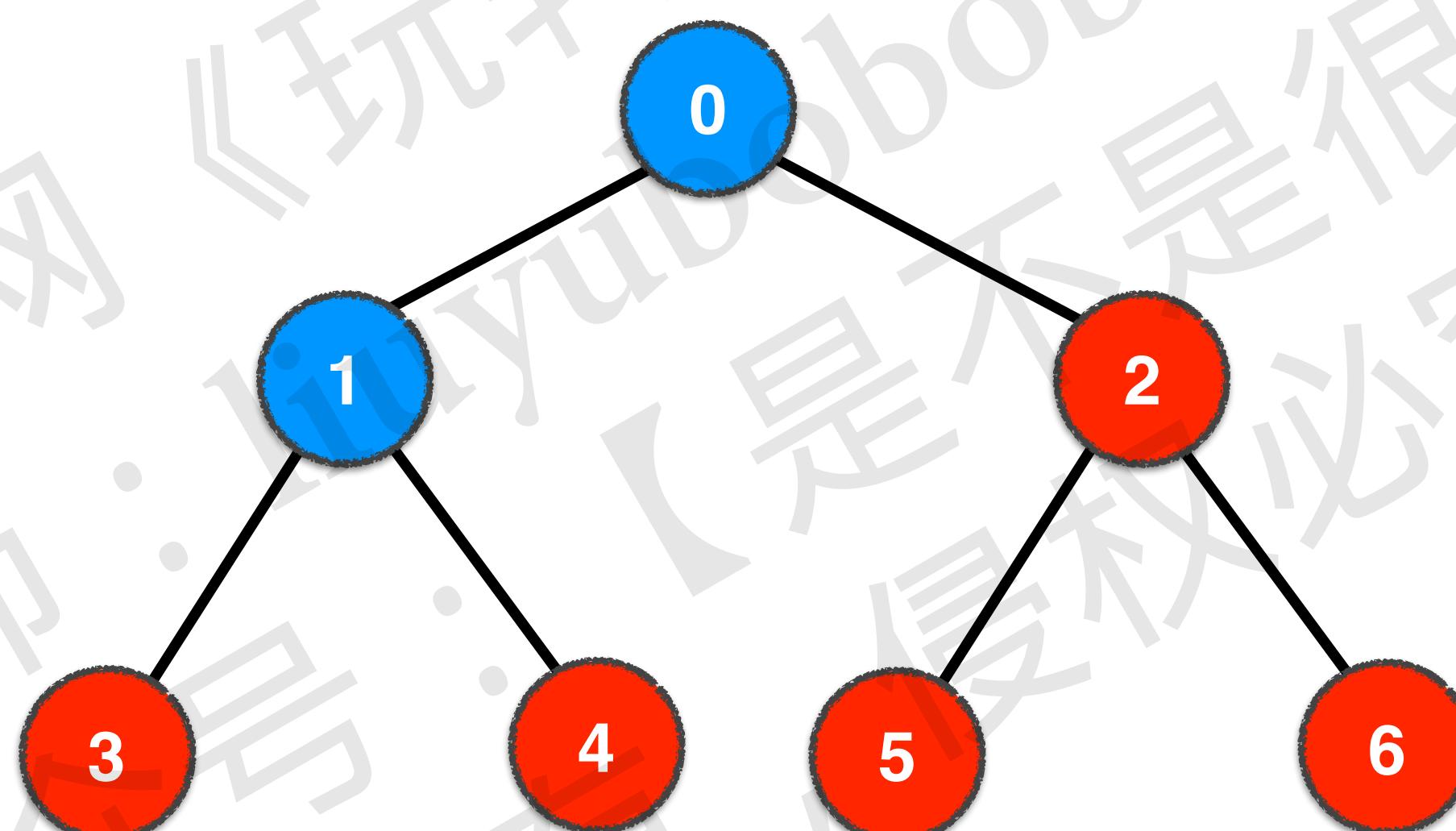
回顾树的广度优先遍历



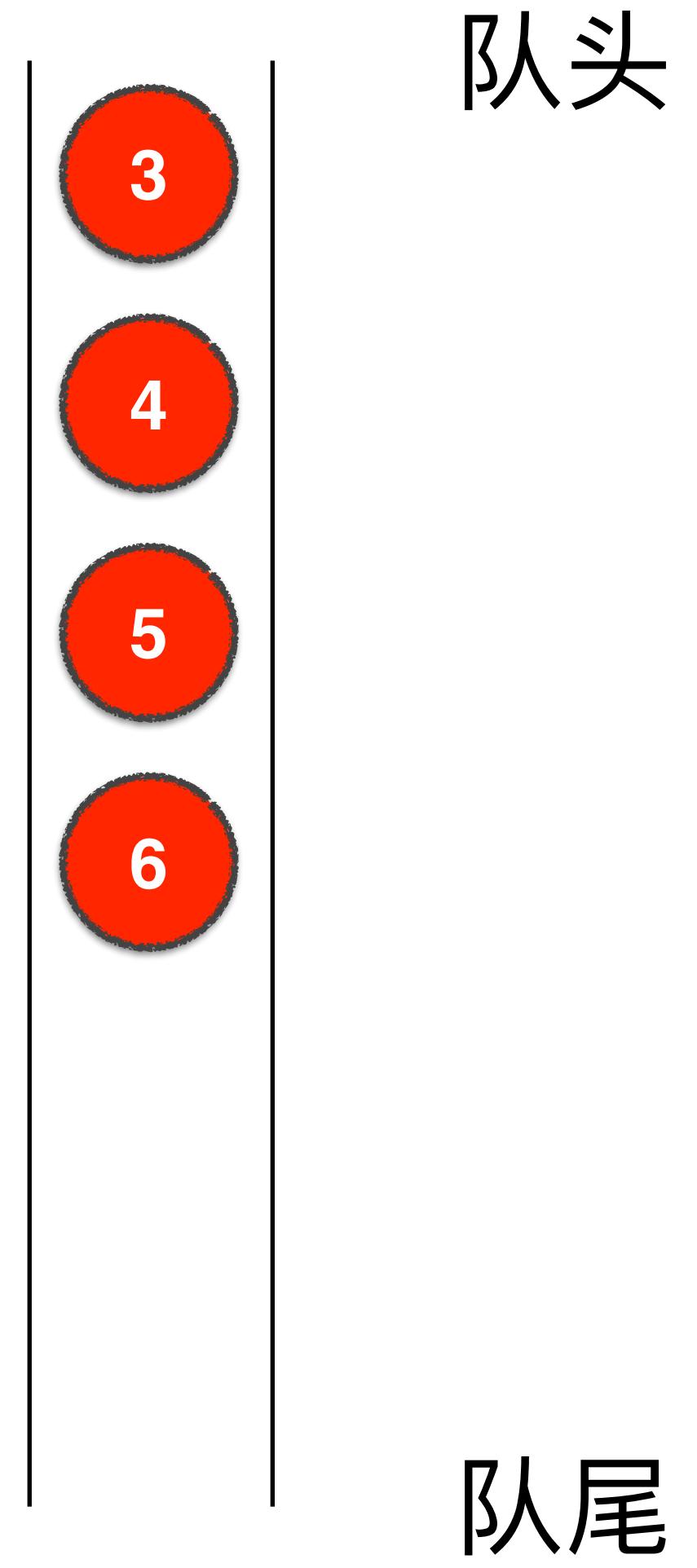
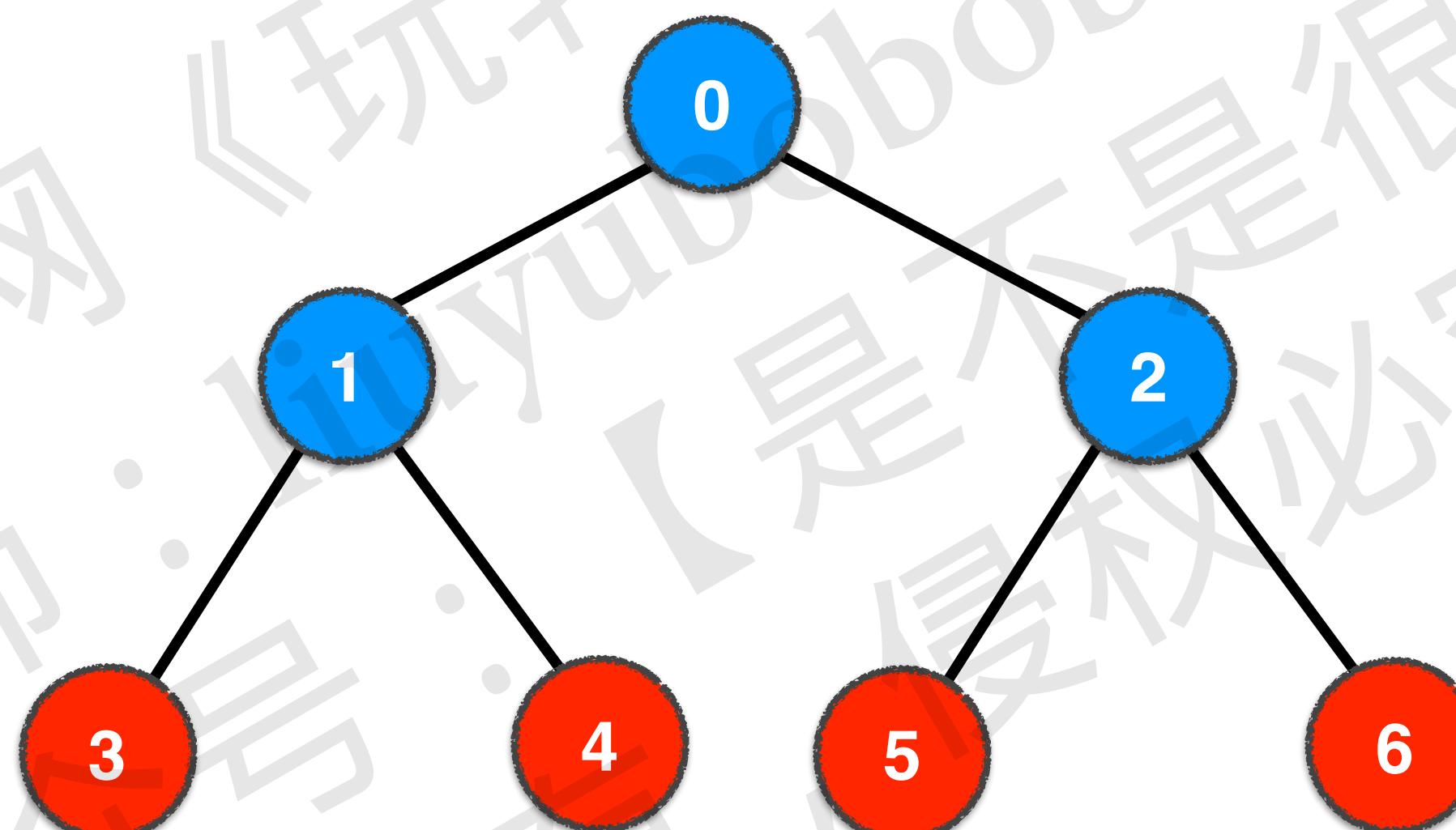
回顾树的广度优先遍历



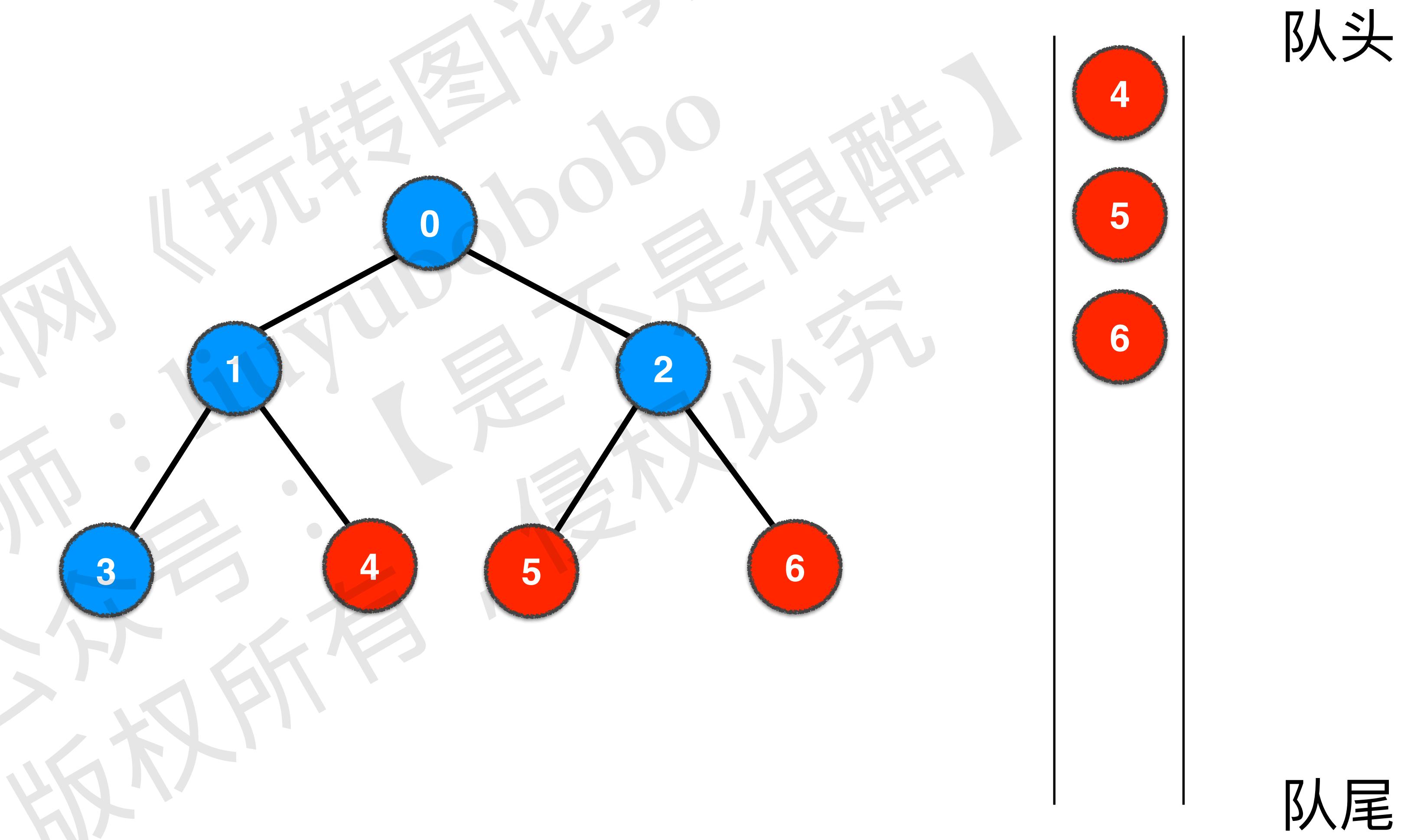
回顾树的广度优先遍历



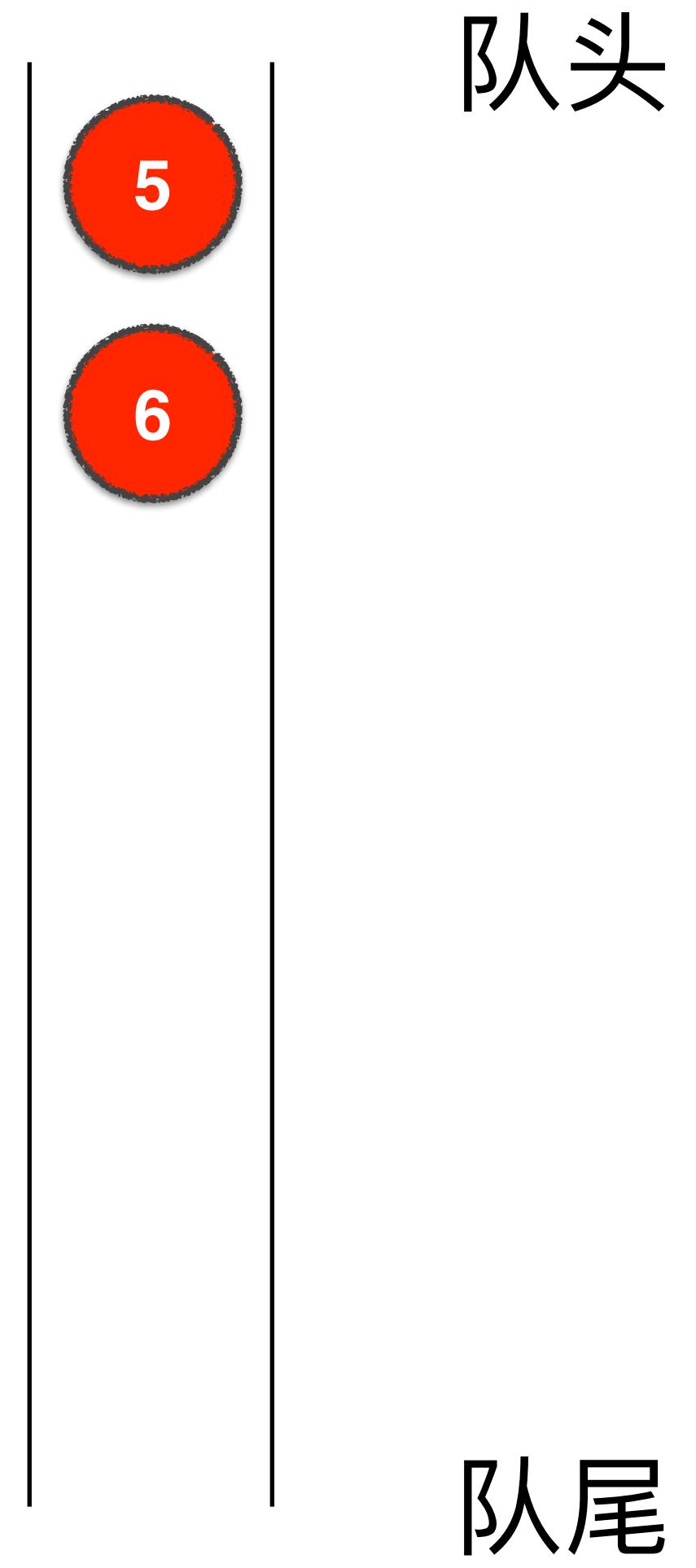
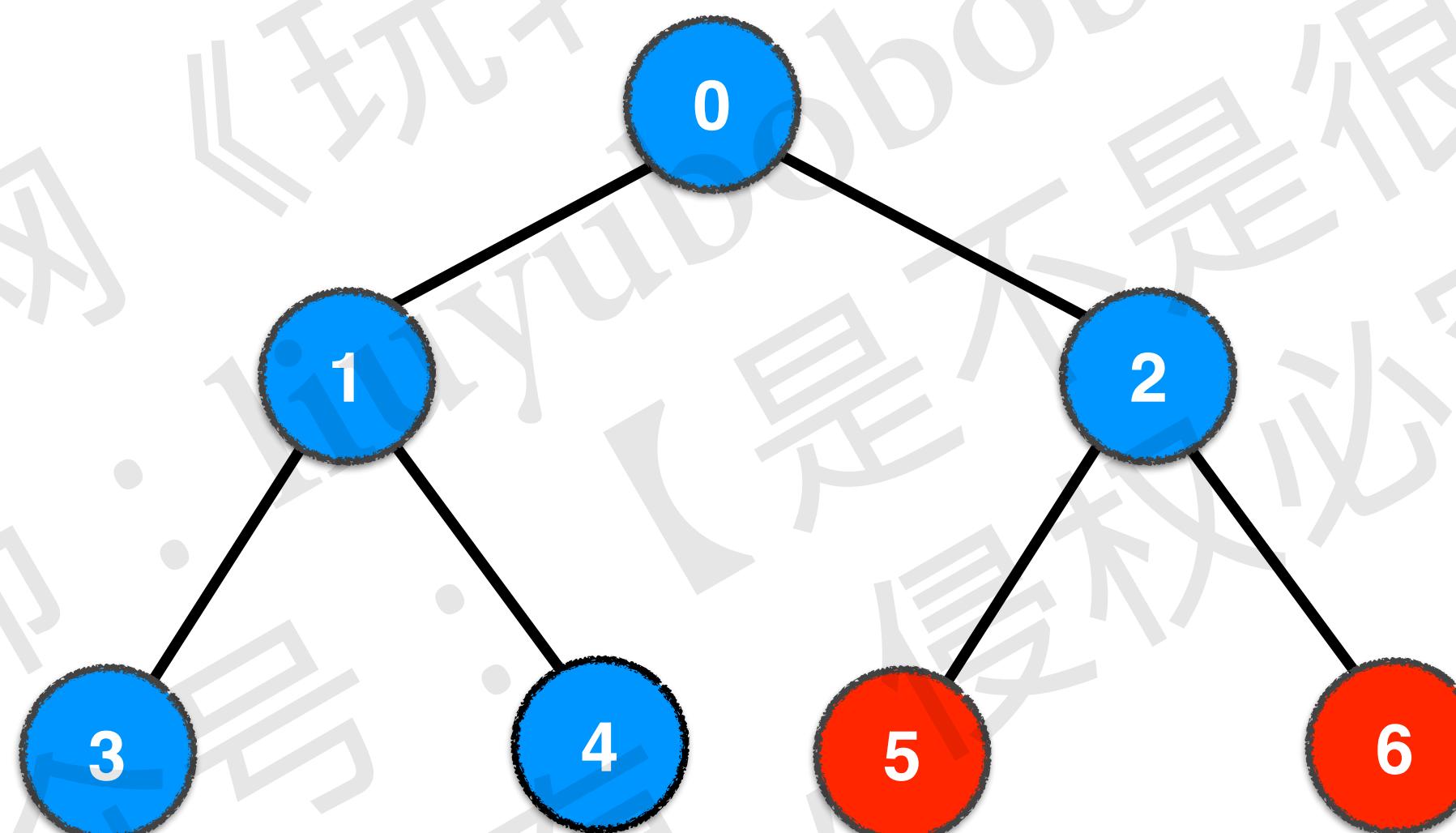
回顾树的广度优先遍历



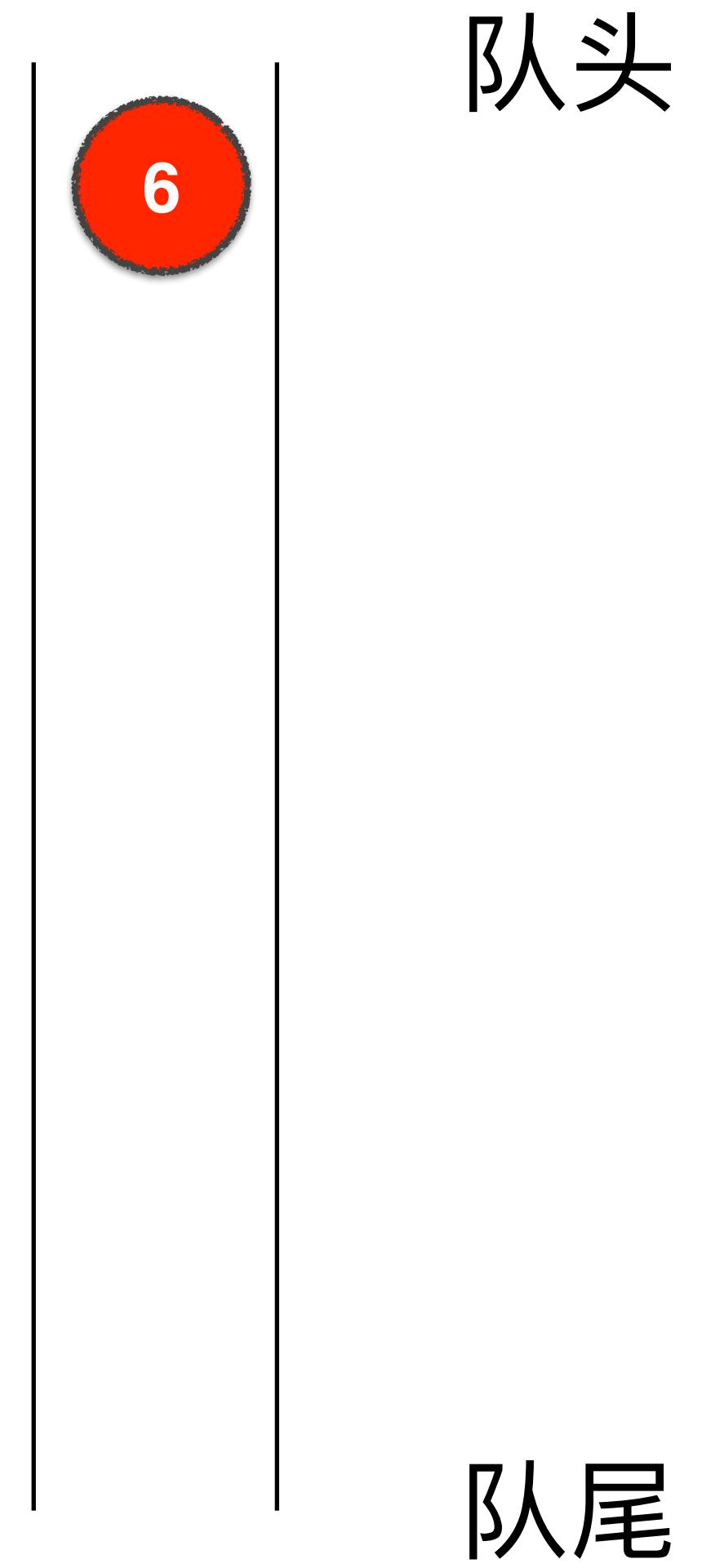
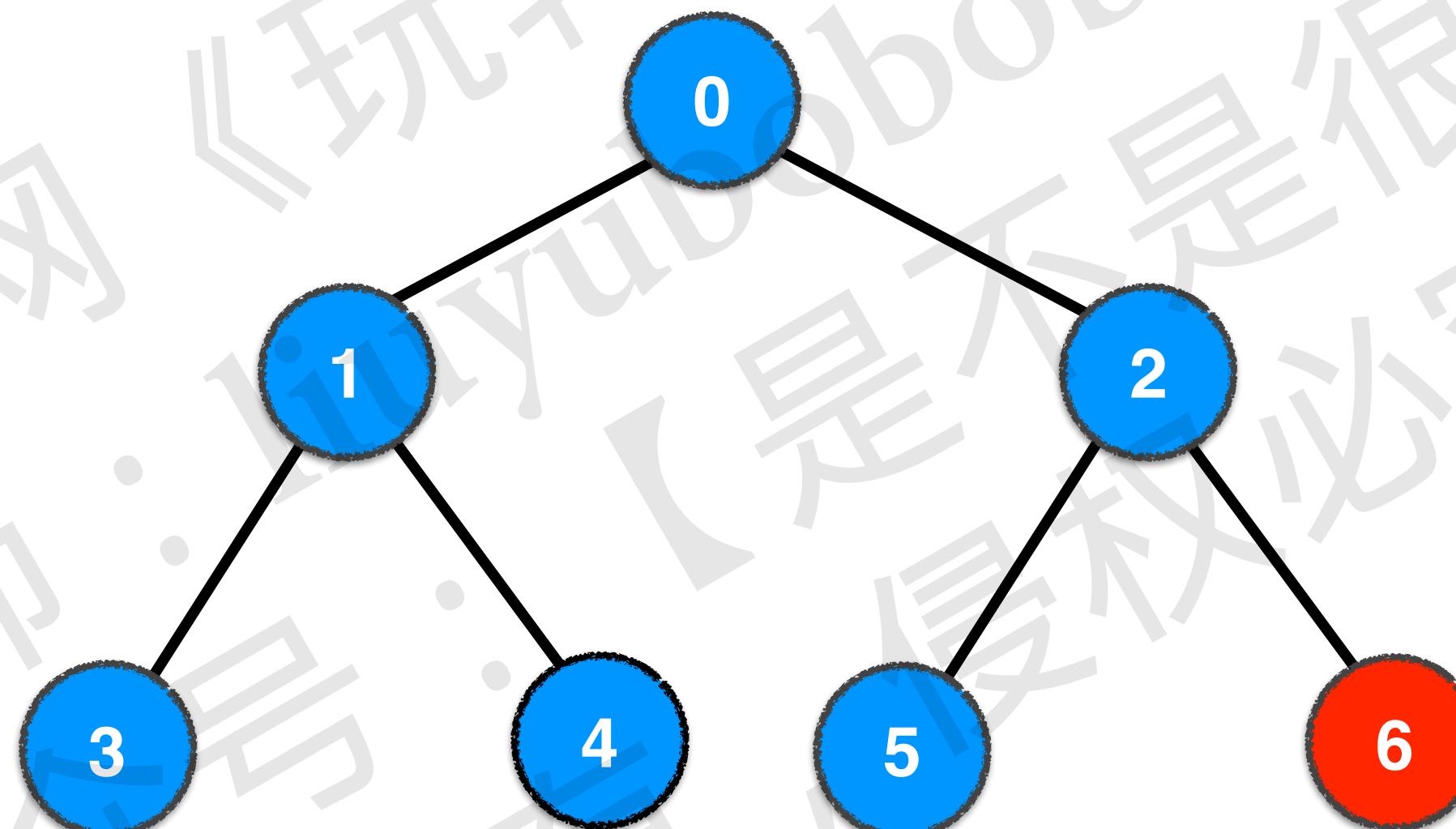
回顾树的广度优先遍历



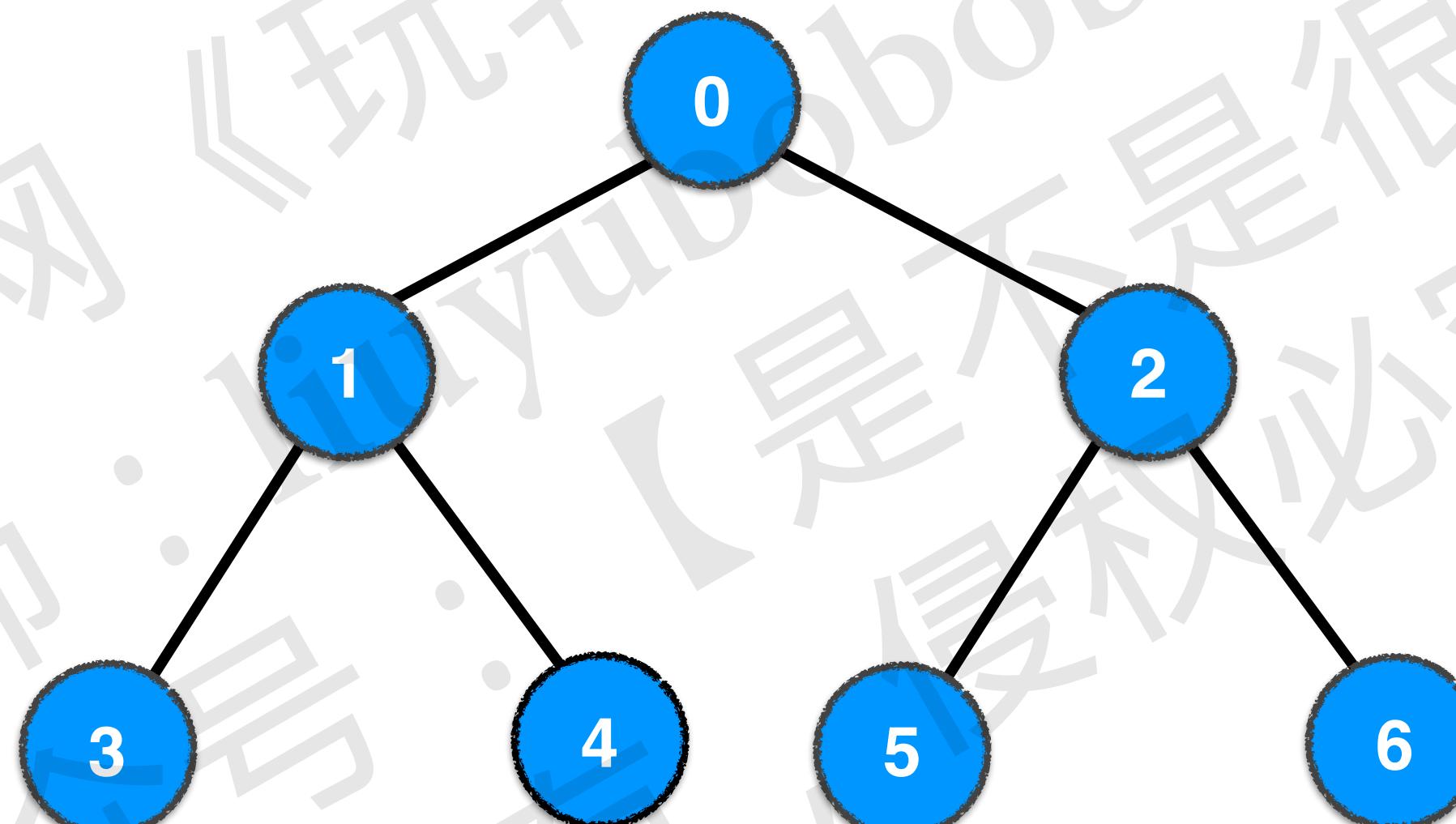
回顾树的广度优先遍历



回顾树的广度优先遍历



回顾树的广度优先遍历



队头

队尾

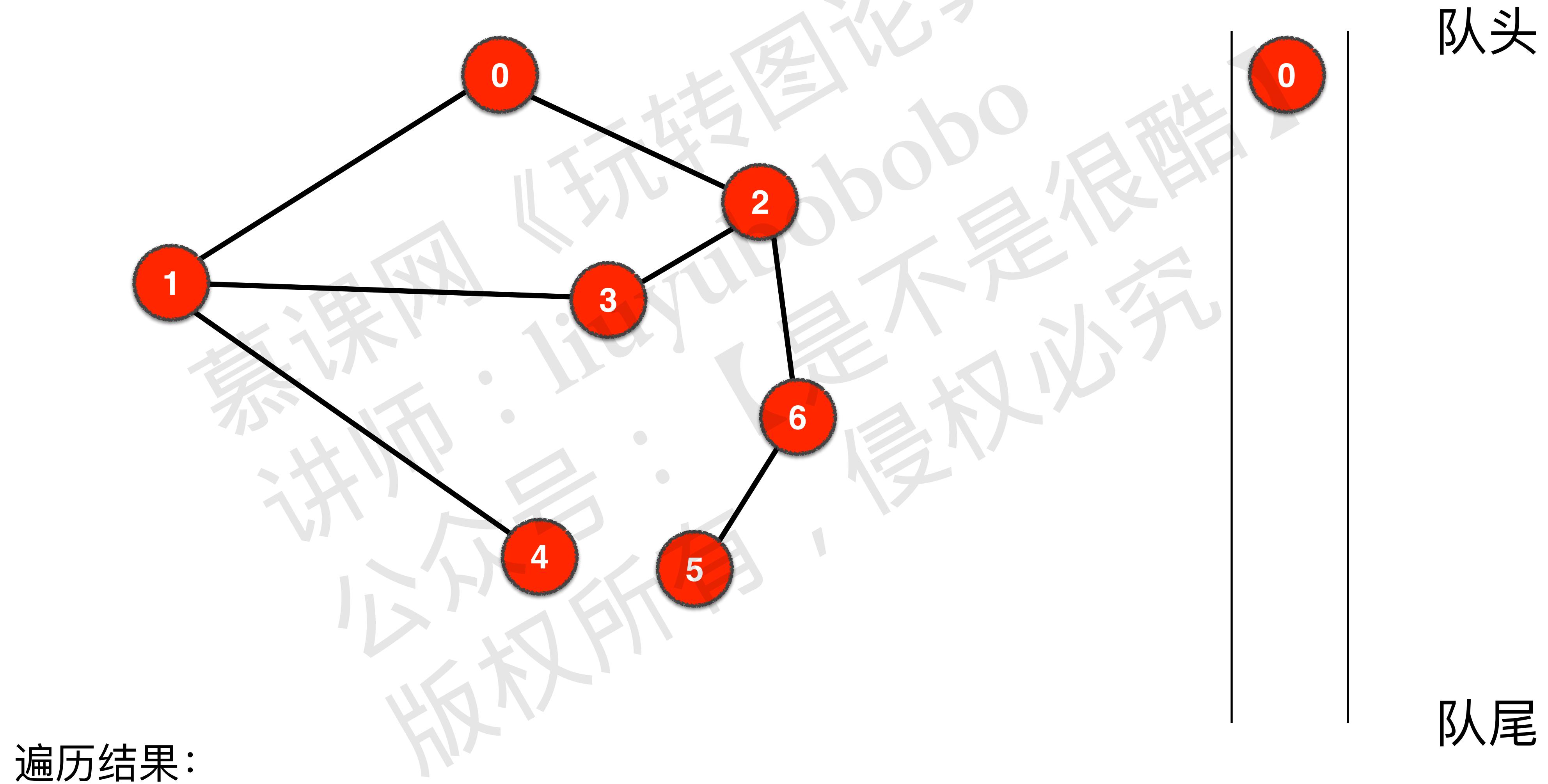
回顾树的广度优先遍历

图的广度优先遍历：完全一样！

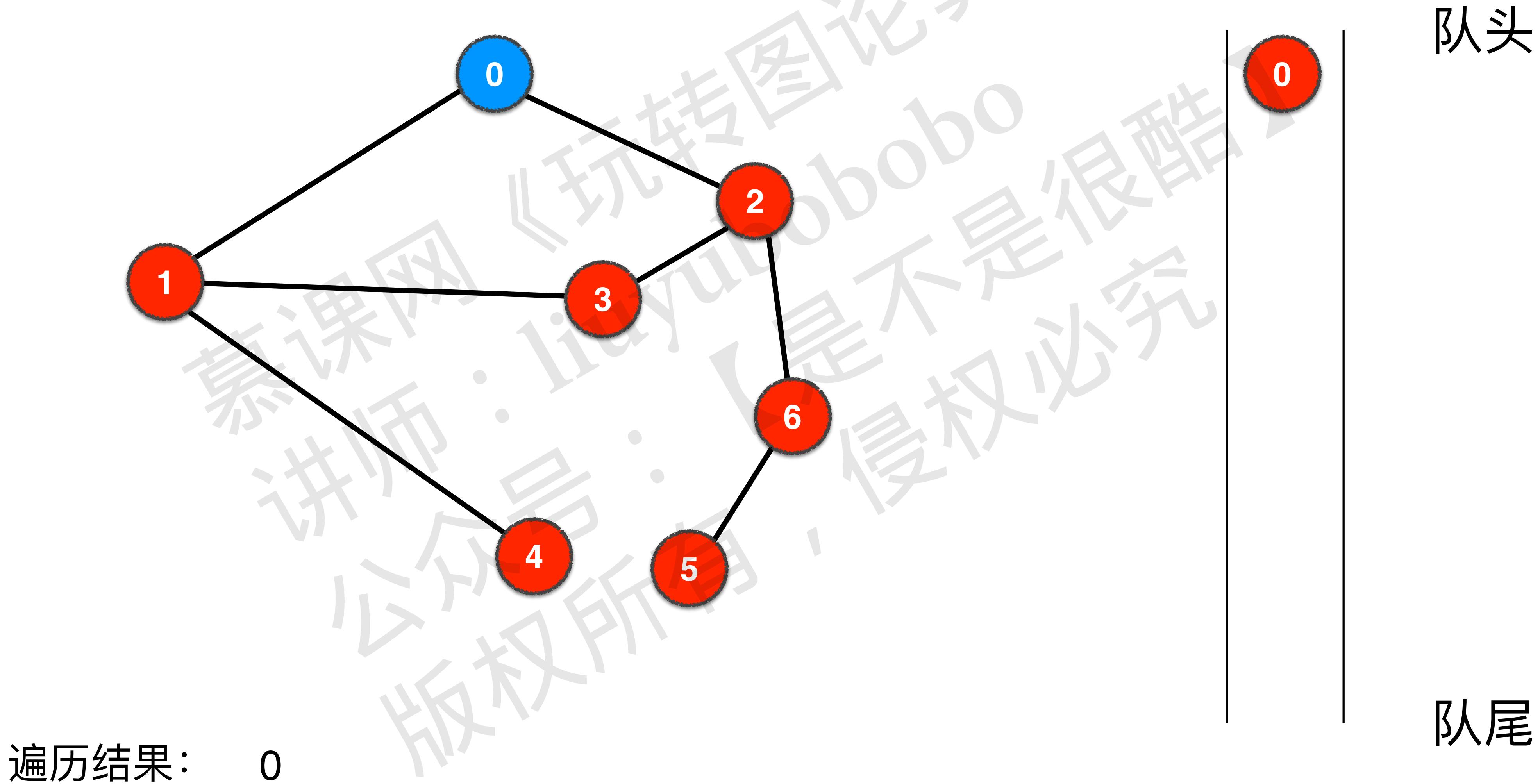
区别：和深度优先遍历一样，
需要记录每个节点是否遍历

visited

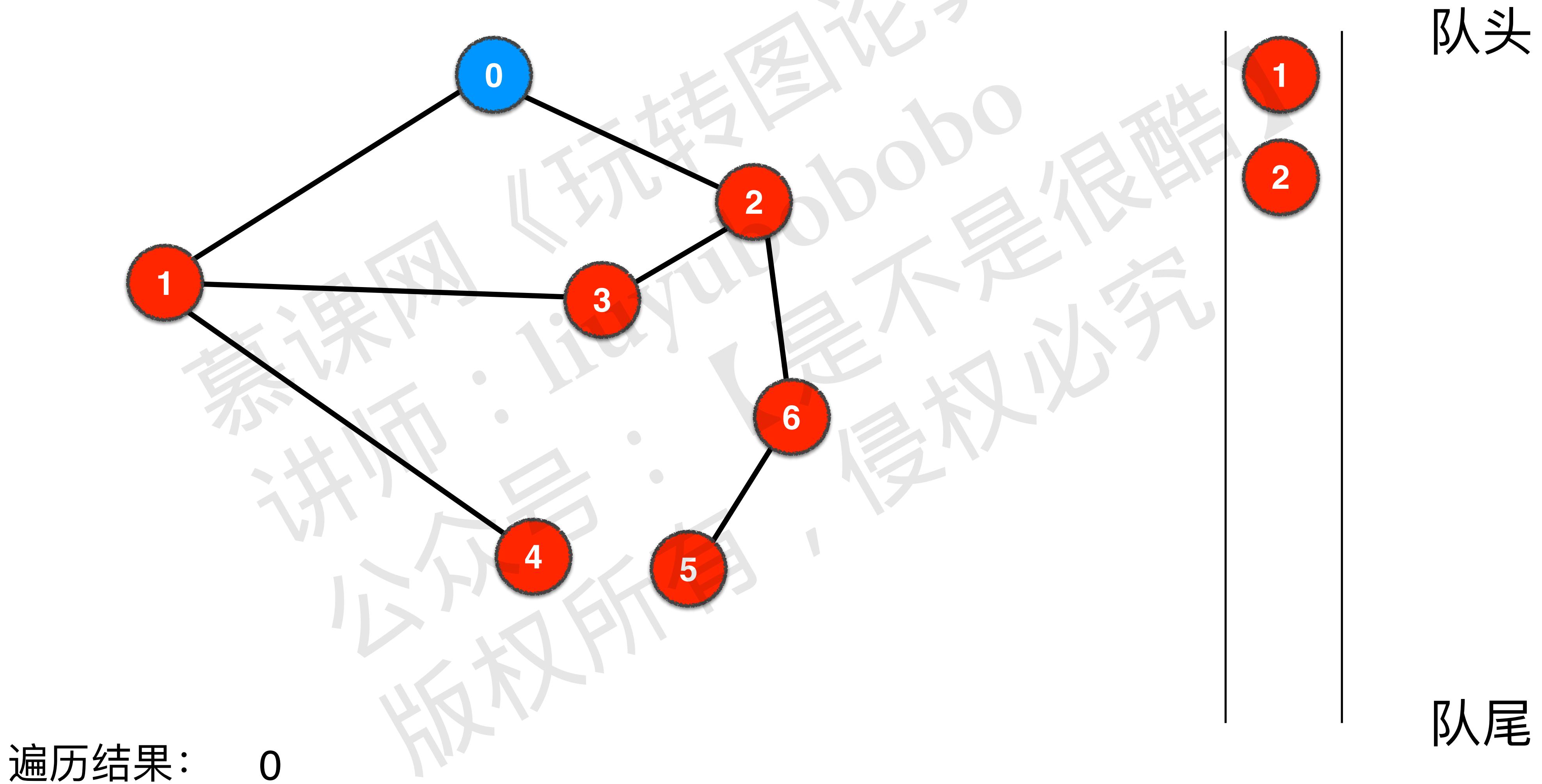
图的广度优先遍历



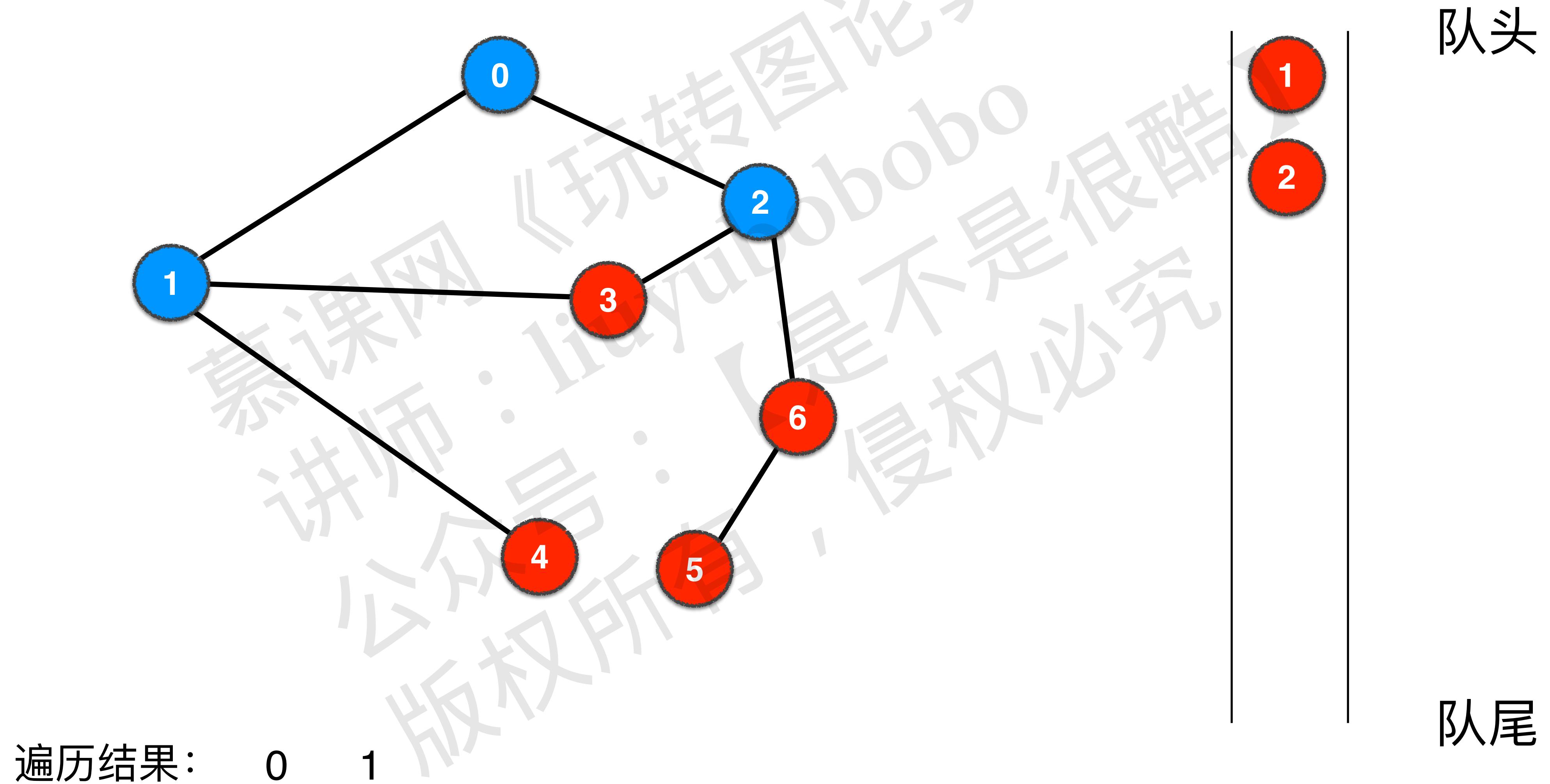
图的广度优先遍历



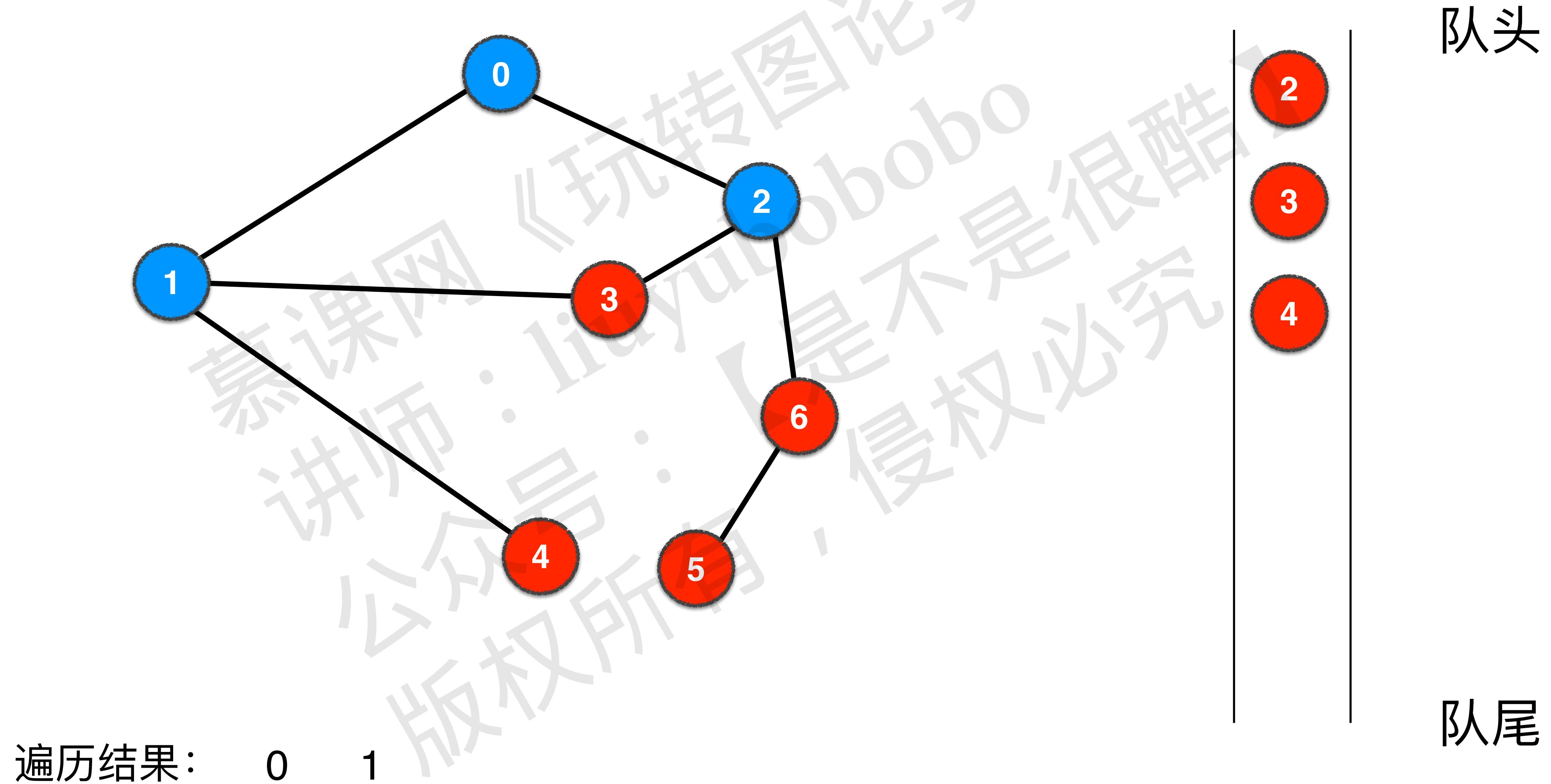
图的广度优先遍历



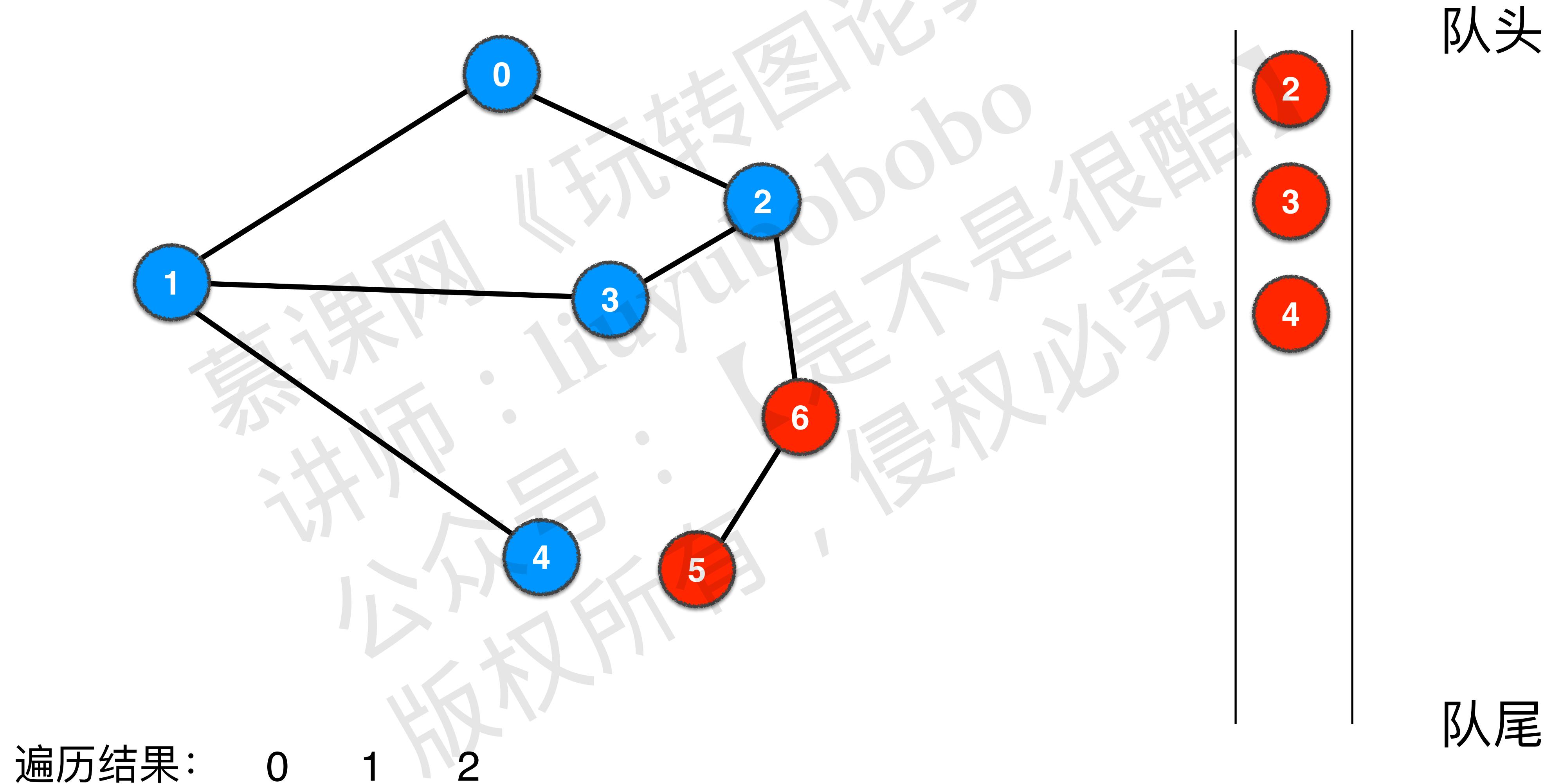
图的广度优先遍历



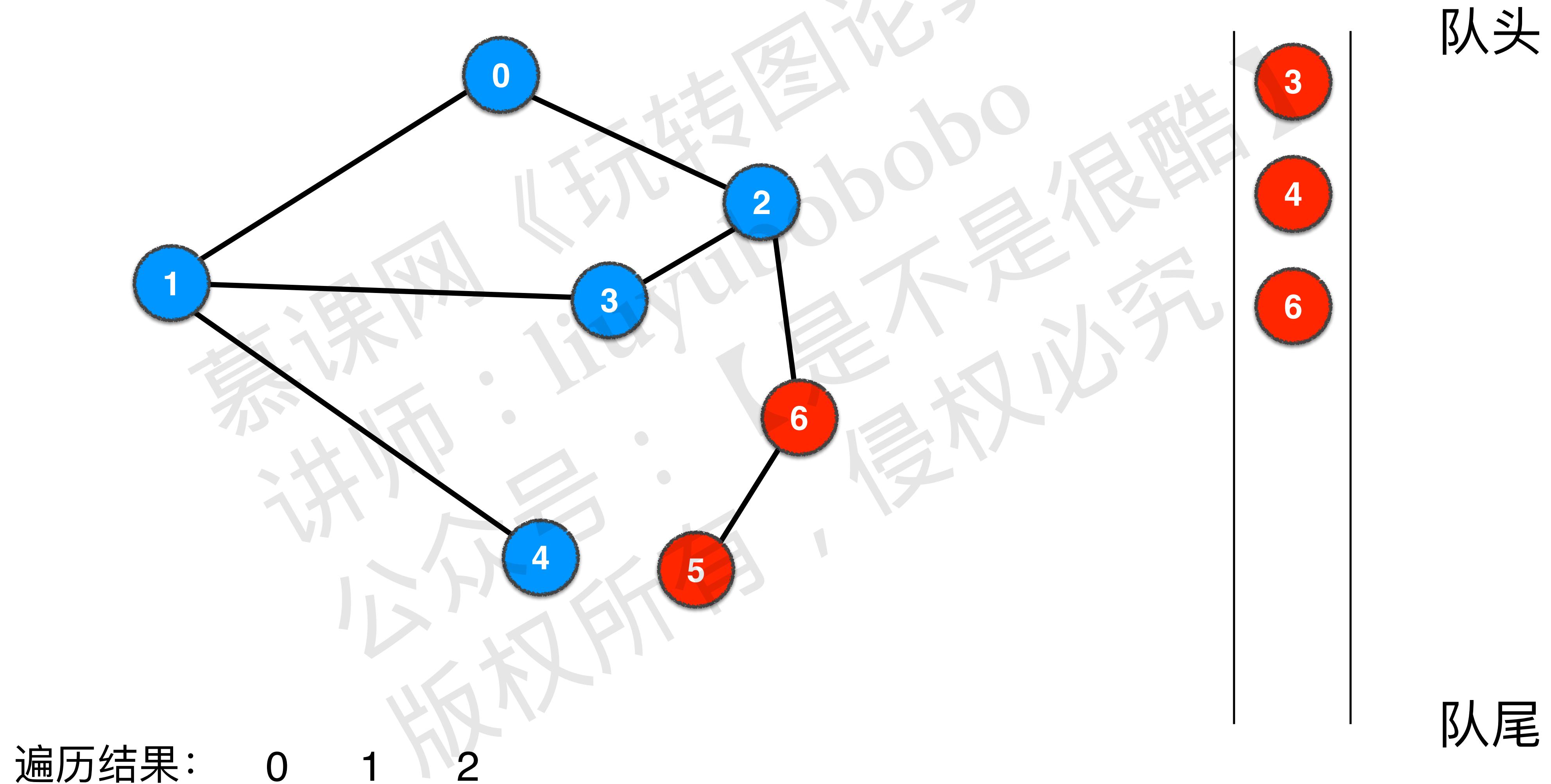
图的广度优先遍历



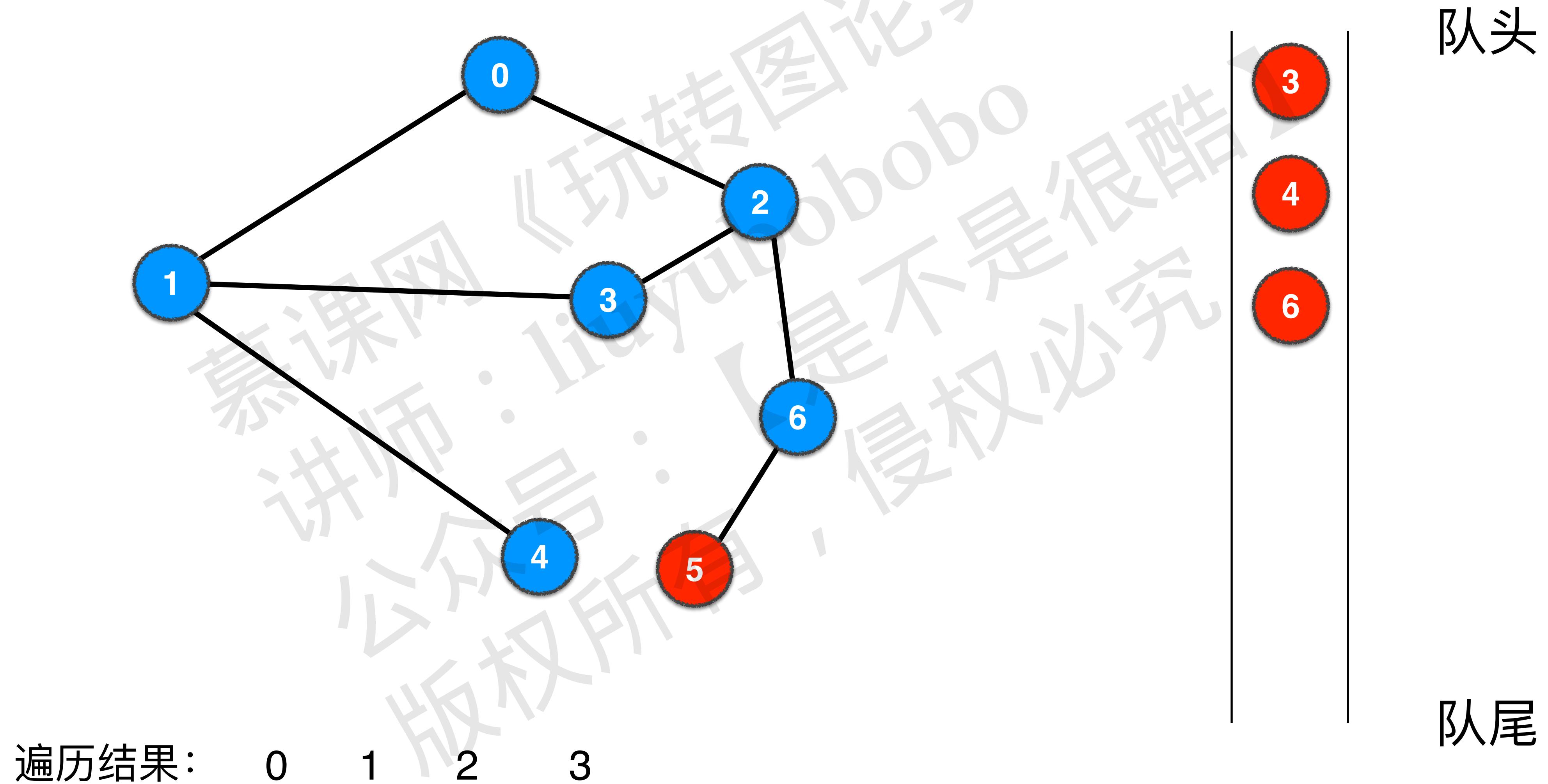
图的广度优先遍历



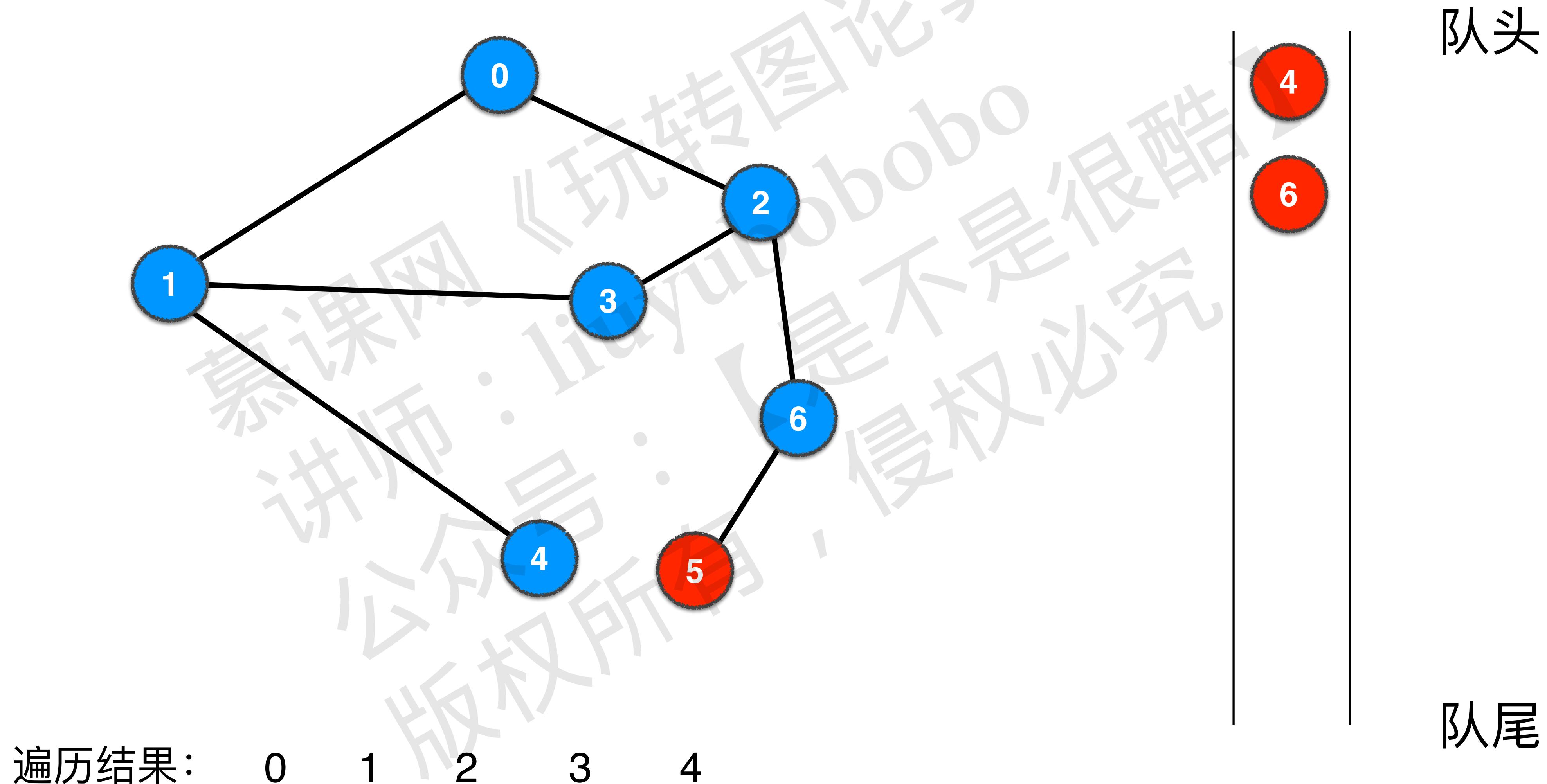
图的广度优先遍历



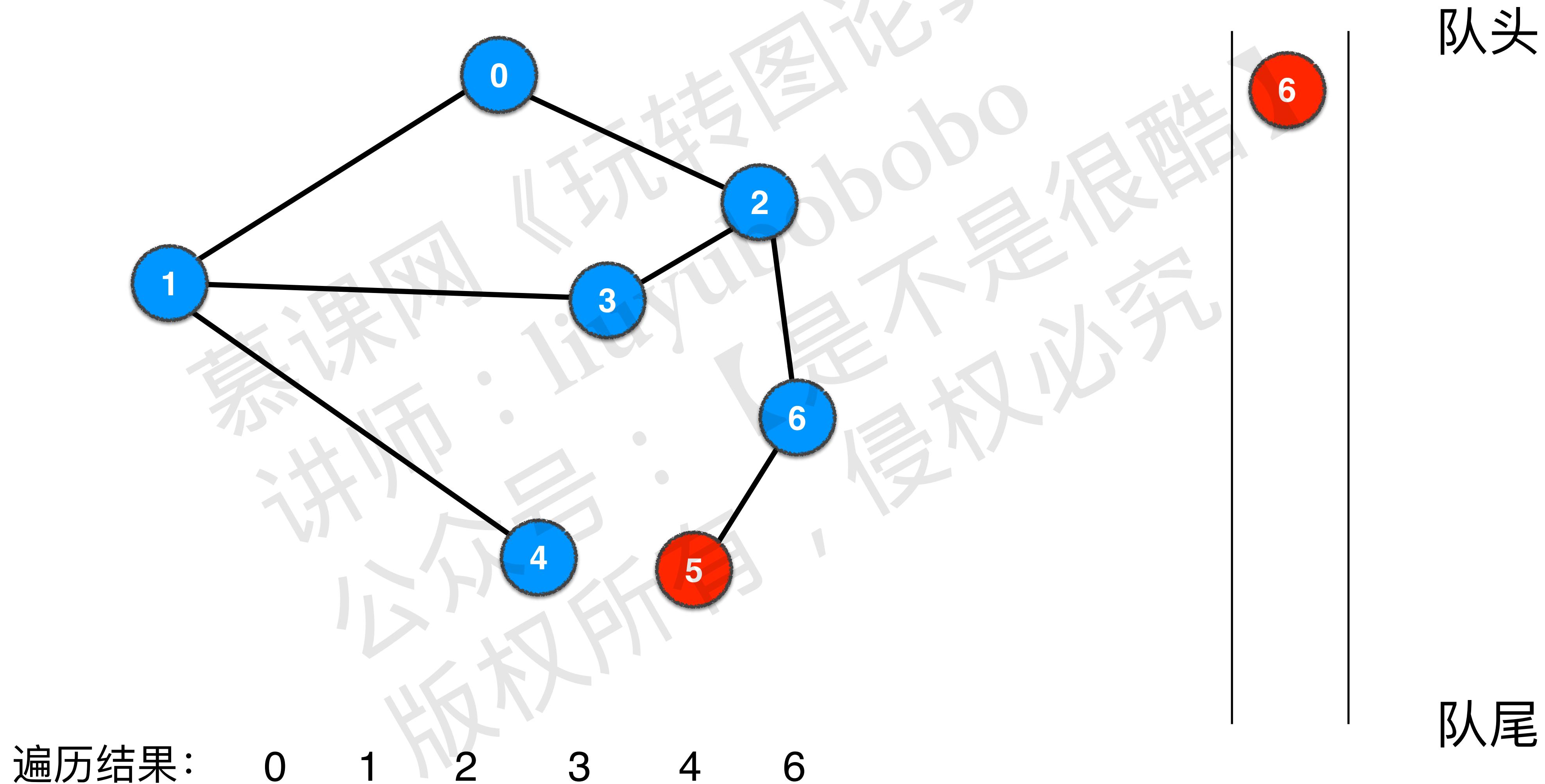
图的广度优先遍历



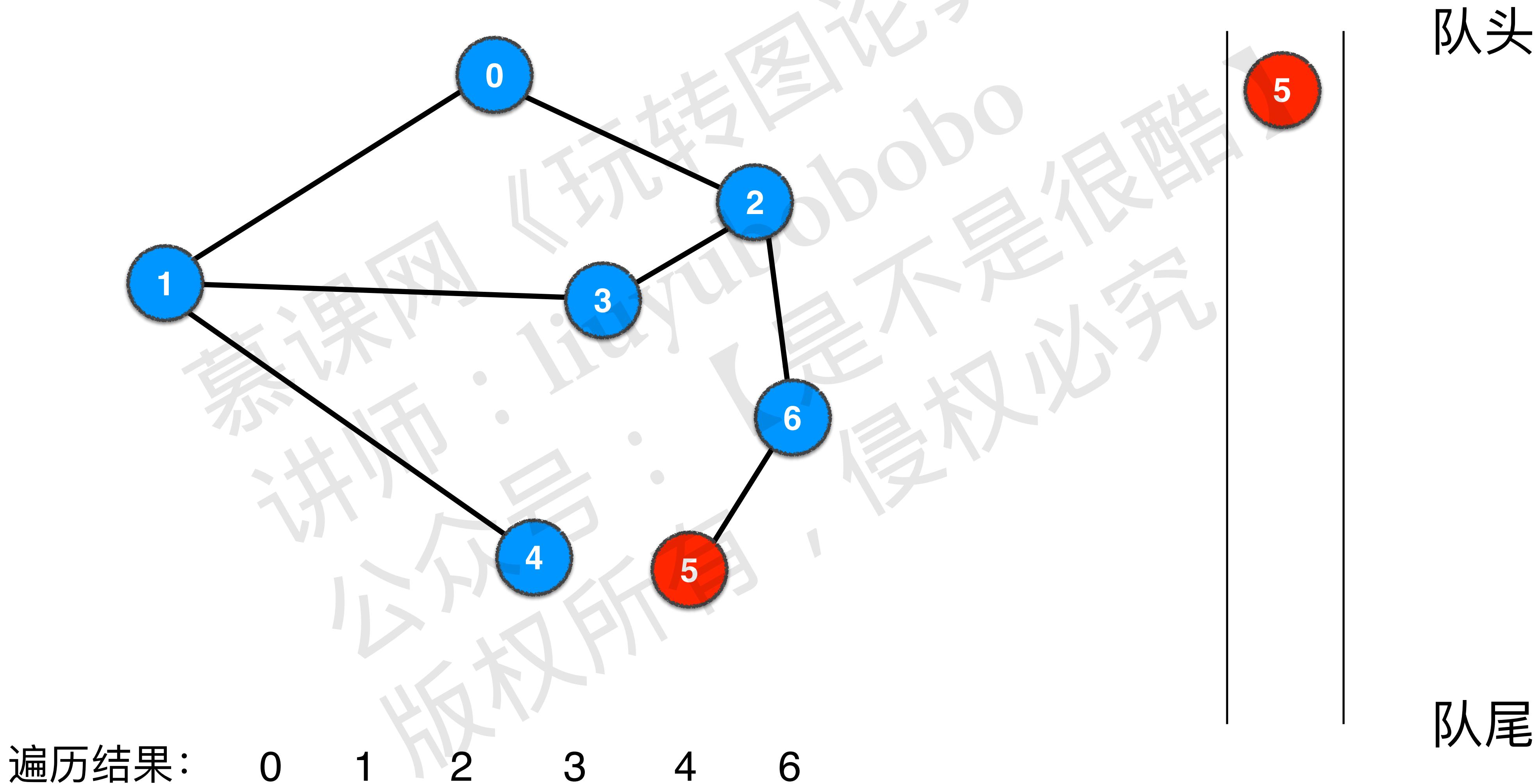
图的广度优先遍历



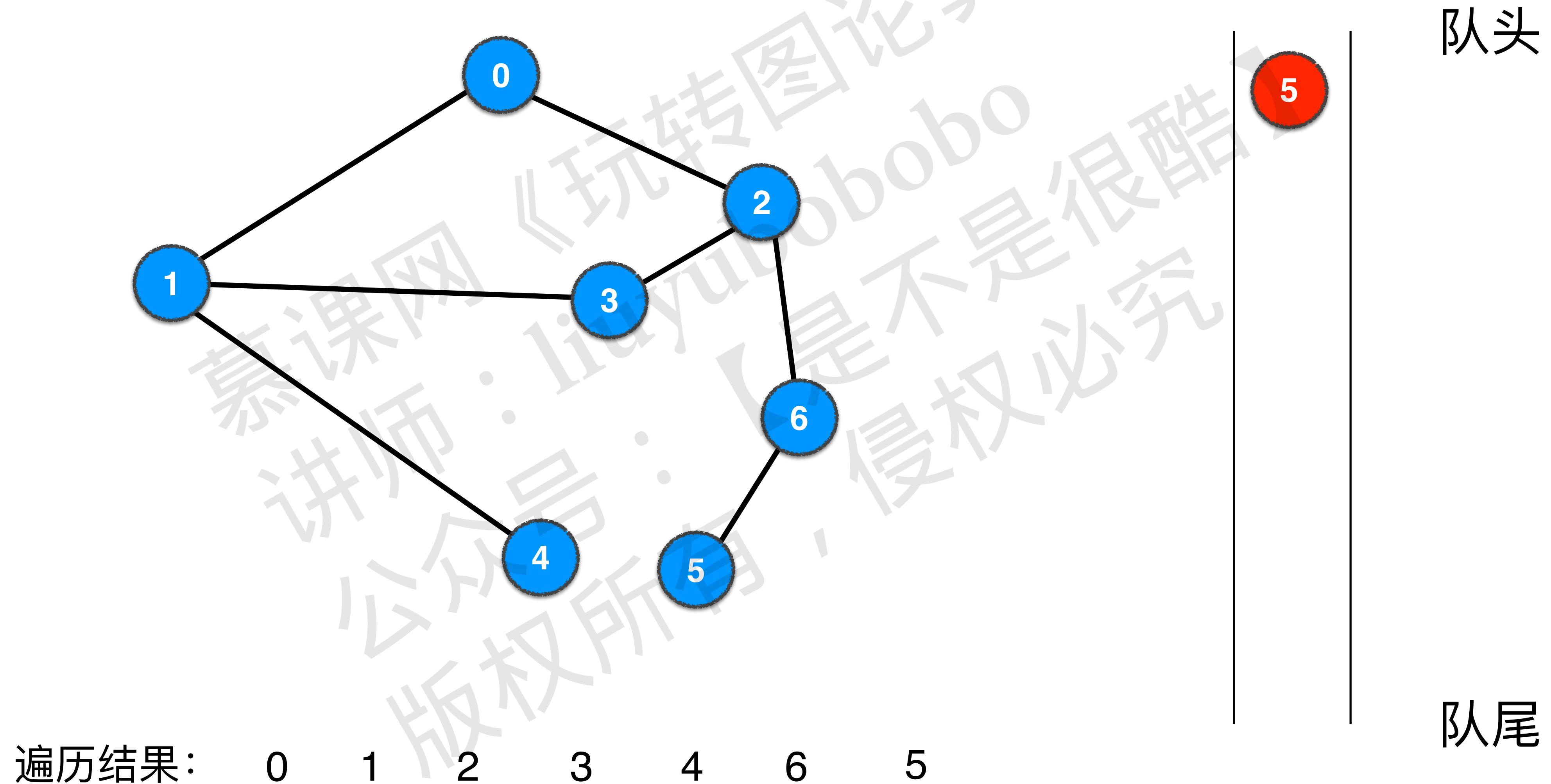
图的广度优先遍历



图的广度优先遍历



图的广度优先遍历



实现图的广度优先遍历

liuyubobobo

编程实践：实现图的广度优先遍历

慕课网 · liuyan · 《玩转图论算法》

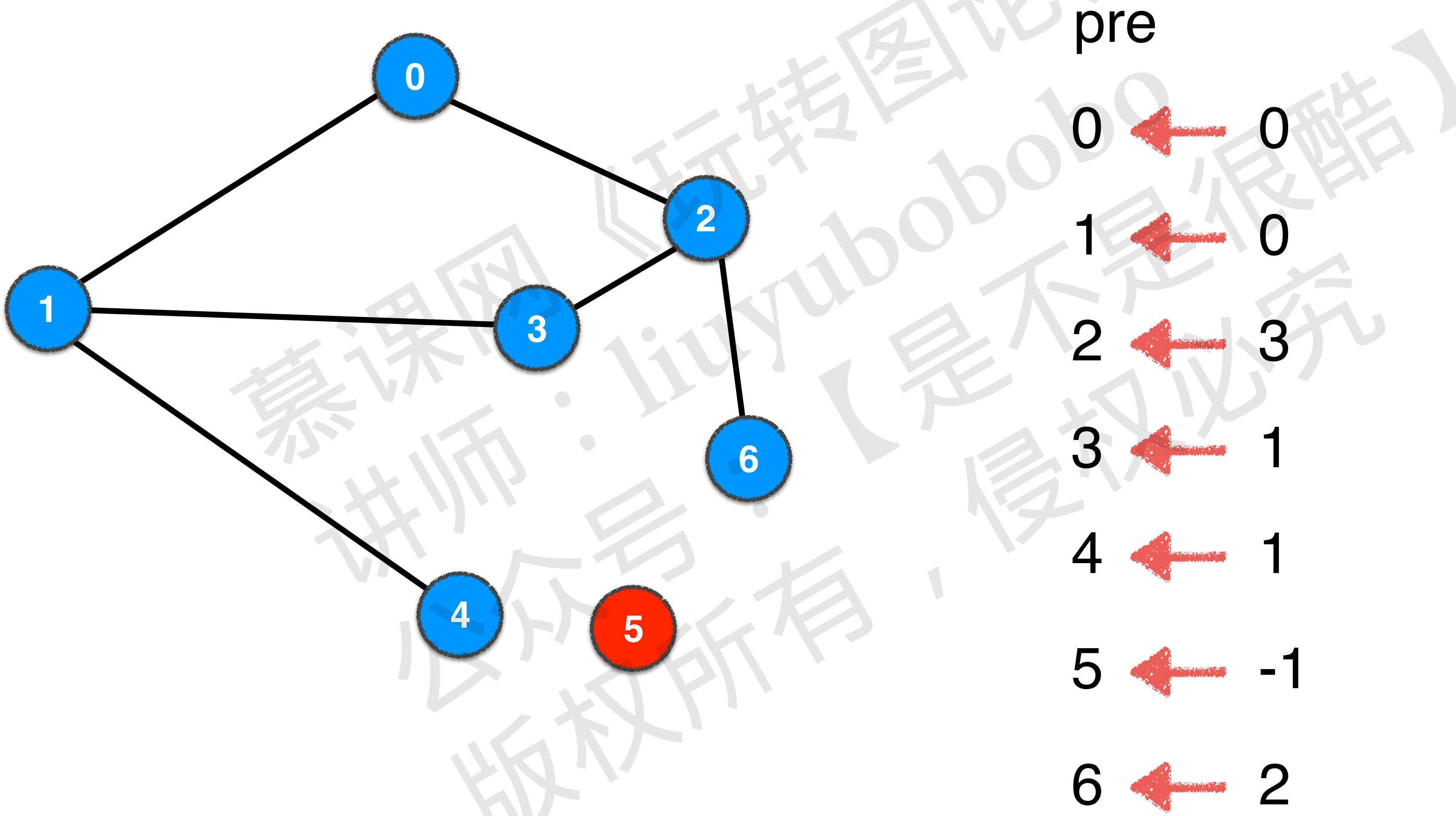
图的广度优先遍历

复杂度： $O(V + E)$

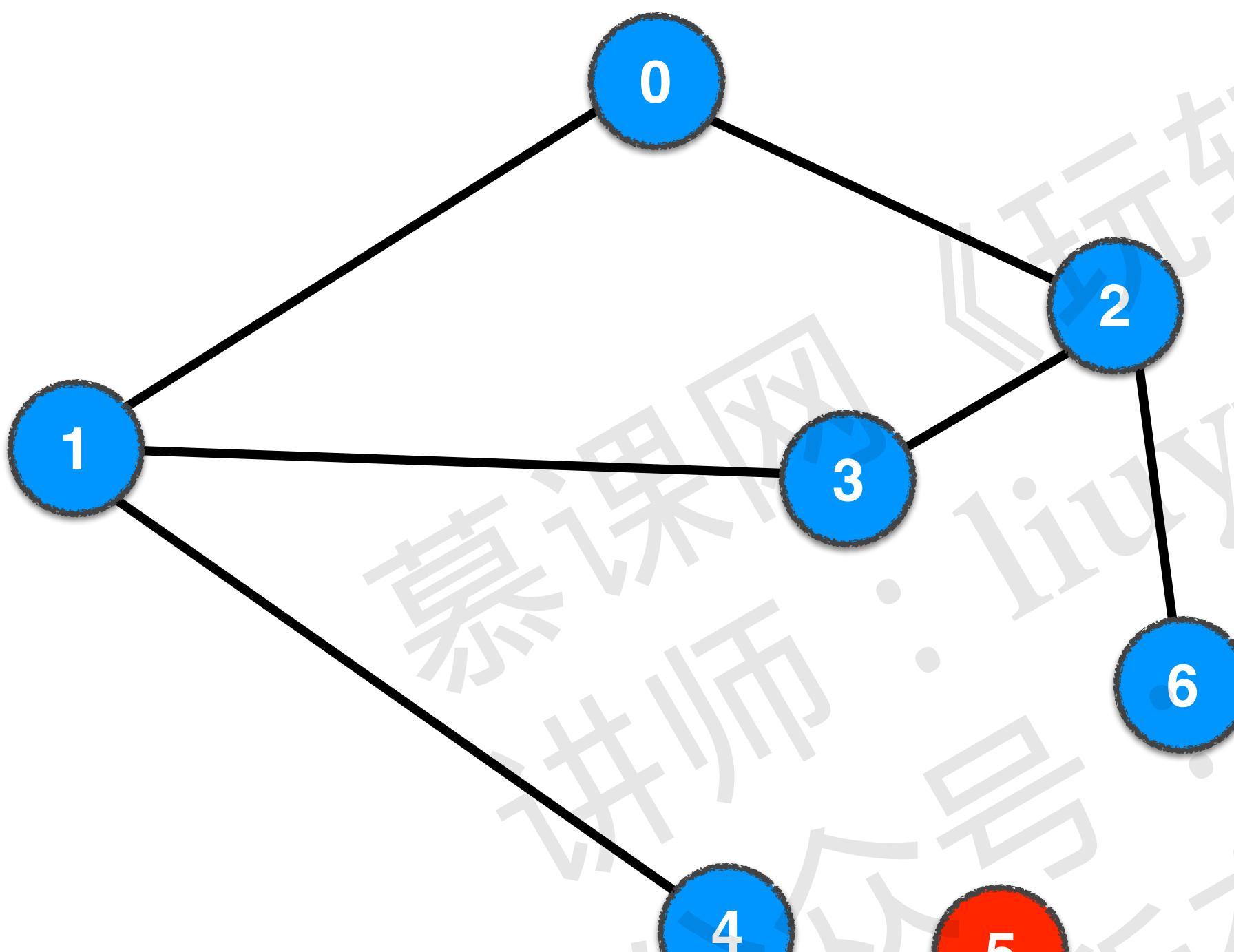
使用BFS求解单源路径问题

liuyubobobo

路径问题



路径问题



```
visited[0...V-1] = false;  
pre[0...V-1] = -1;  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        bfs(v);  
  
bfs(int s){  
  
    queue.add(s)  
    visited[s] = true  
    pre[s] = s;  
    while(!q.isEmpty()){  
        int v = q.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                queue.add(w);  
                visited[w] = true;  
                pre[w] = v;  
            }  
    }  
}
```

编程实践：使用BFS解决单源路径问题

慕课网 · LiuYi · 《玩转图论算法》

路径问题

单源路径问题

所有点对路径问题

提前返回?

其他BFS应用

联通分量 (个数, 具体联通分量)

环检测

二分图检测

更多使用BFS求解路径问题

liuyubobobo

文字：更多使用BFS求解路径问题

慕课网 · LiuYan · 《玩转图论算法》
讲师 · bobobo · 『最炫必修课』
版权归公众所有

使用BFS求解联通分量

liuyubobobo

文字：使用BFS求解联通分量

慕课师 · LiuYJ · 《图论与算法》 · 课件 · 公众所有 · 版权所有 · 严禁盗用 · 请勿商用 · 侵权必究 · 《玩转图论算法法》

使用BFS求解环检测

liuyubobobo

文字：使用BFS求解环检测

慕课网 · liuyan · 《玩转图论算法》

使用BFS求解二分图检测

liuyubobobo

文字：使用BFS求解二分图检测

慕课网《玩转图论算法》
讲师：liuyibobobo
（题权必归讲师所有，侵权必究）

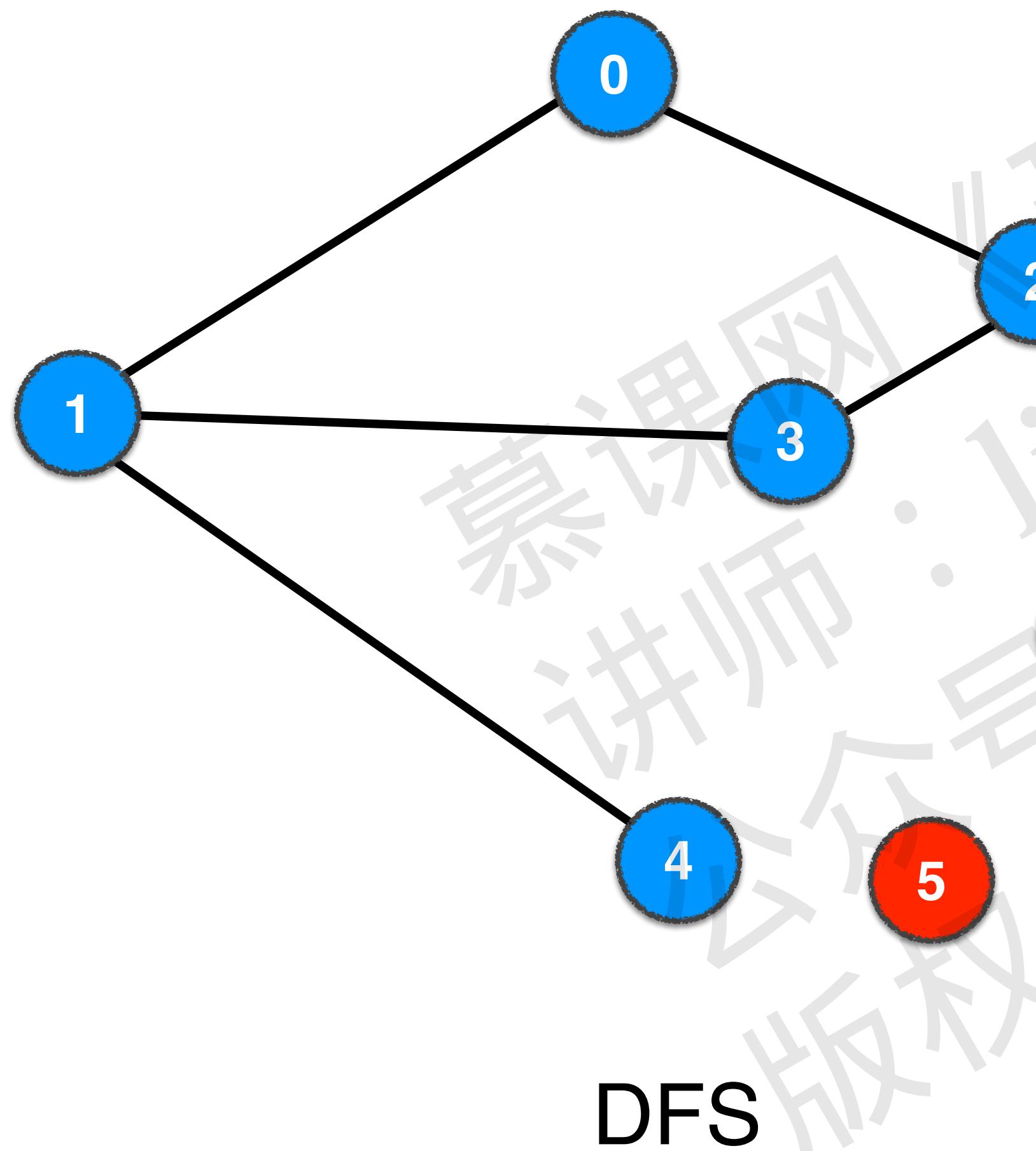
BFS的重要性质

liuyubobobo

观察BFS求出的结果

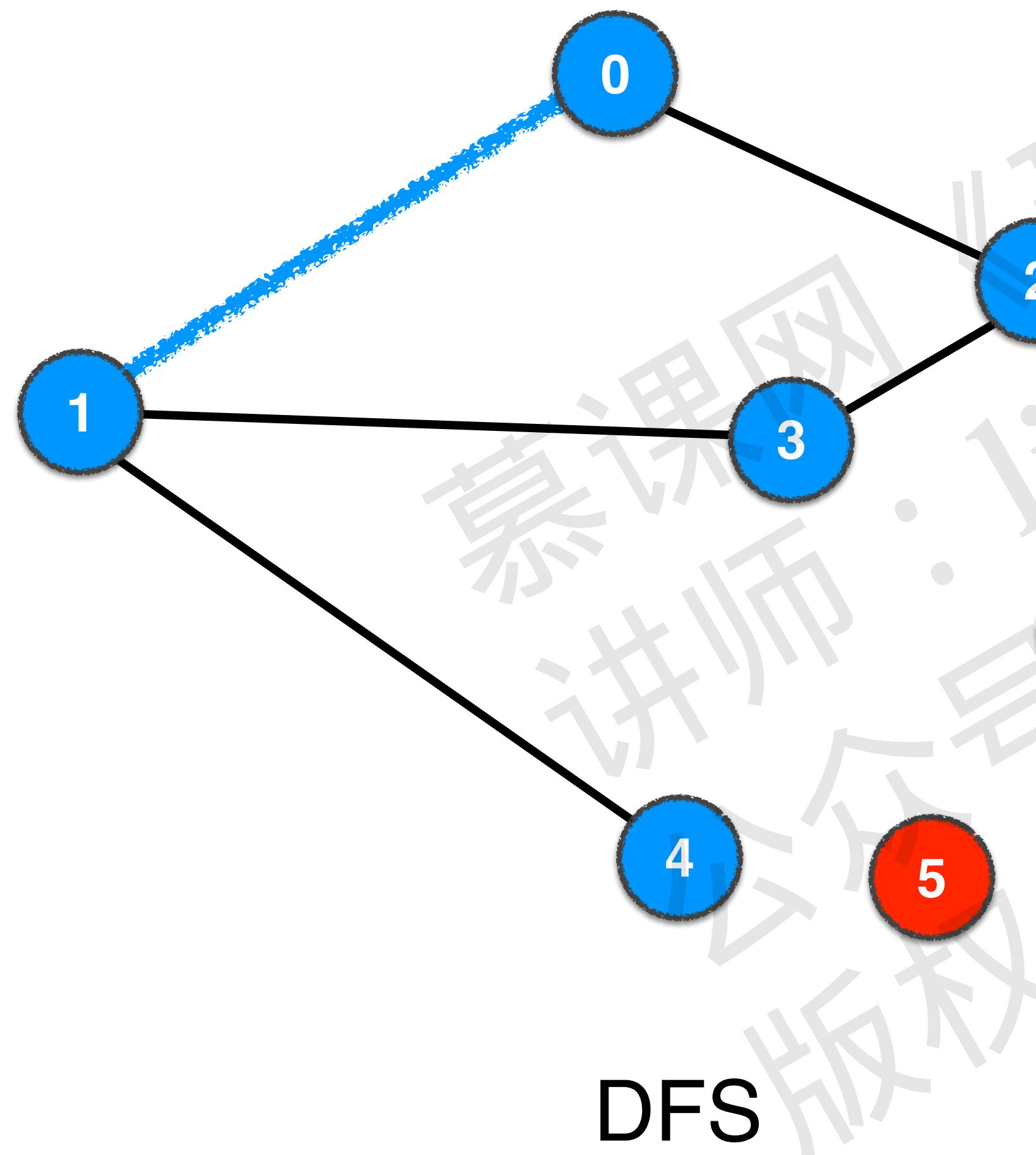
慕课网《玩转图论算法》
讲师：刘波波
版权归慕课网所有，侵权必究

BFS的重要性质



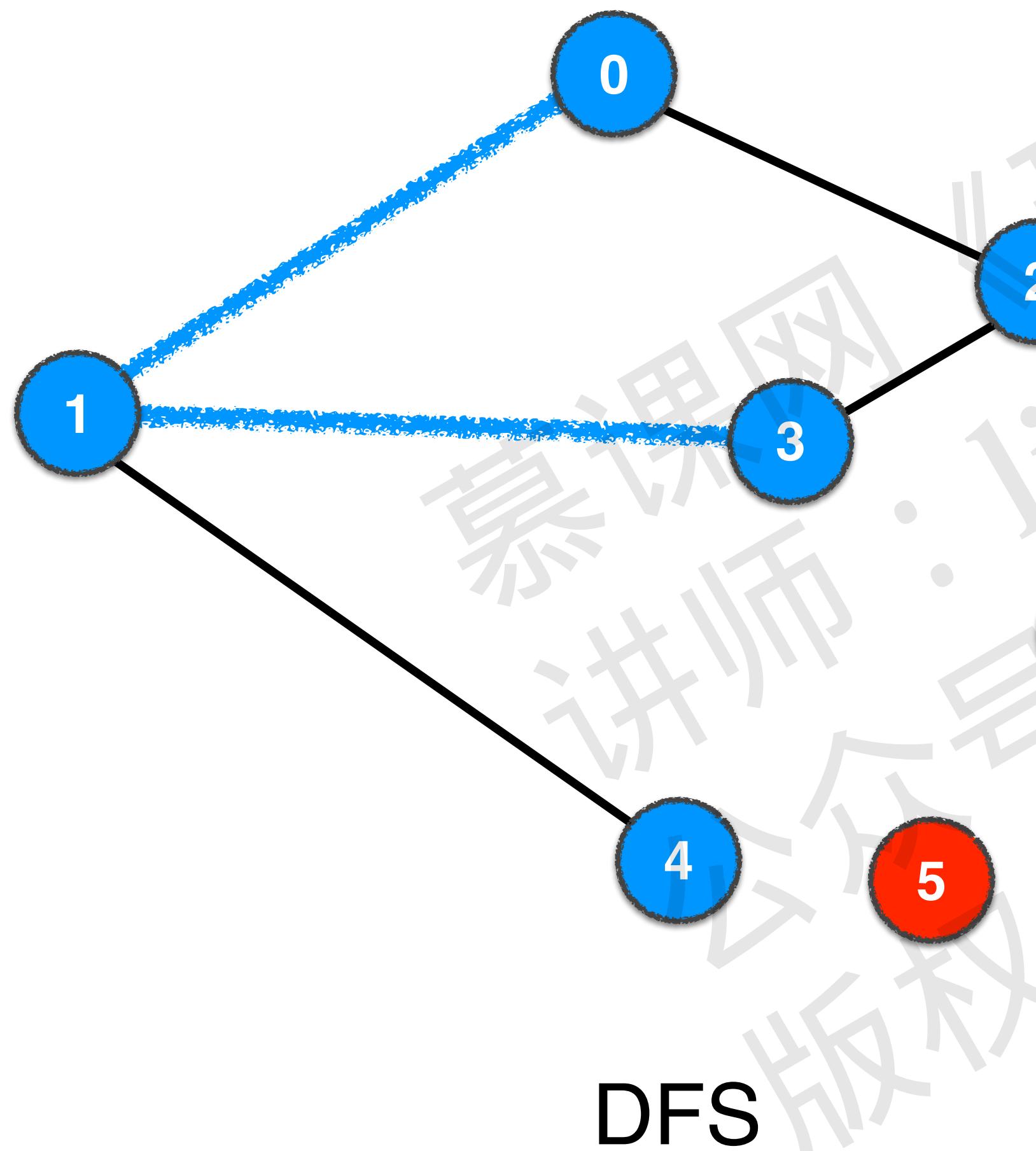
慕课网 / 玩转图论 / 讲师：liuyubobobo / 【是不是很酷】
讲师 / 版权所有，侵权必究

BFS的重要性质



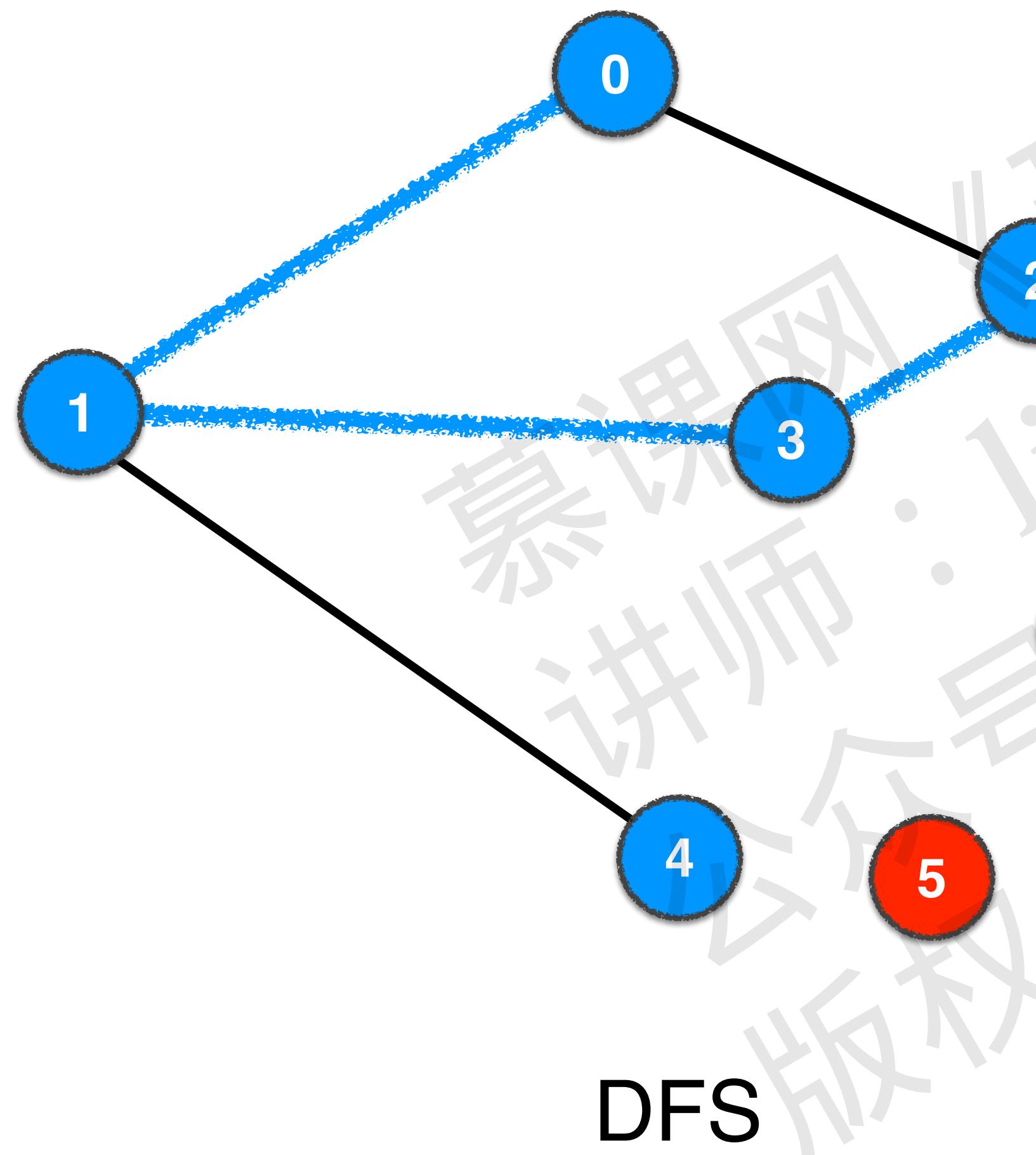
慕课网 / 任铄 / 讲师 / liuyubobobo / 【是不是很酷】
【玩转图论界】 / BFS的重要性质 / 版权所有，侵权必究

BFS的重要性质



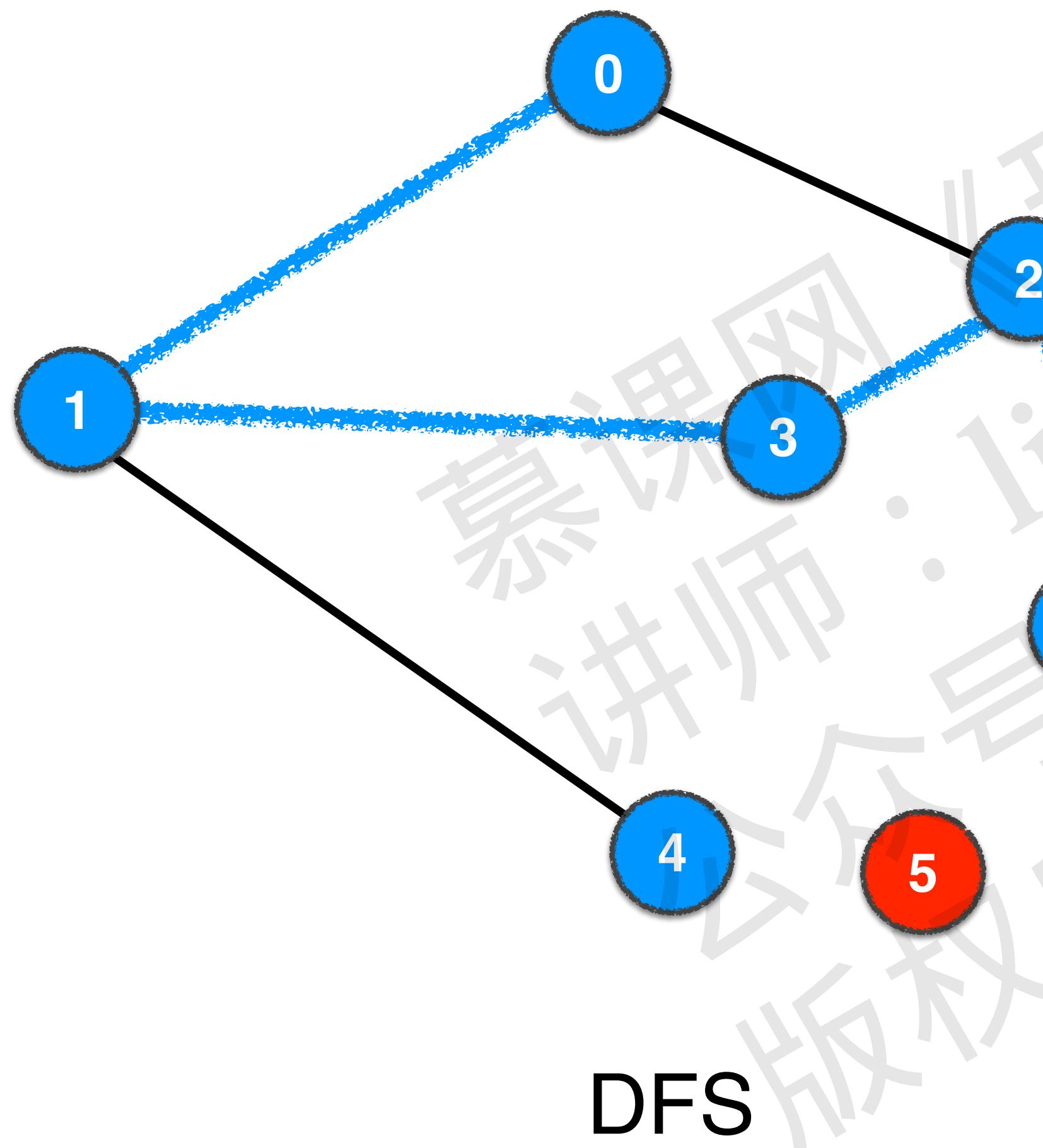
DFS

BFS的重要性质



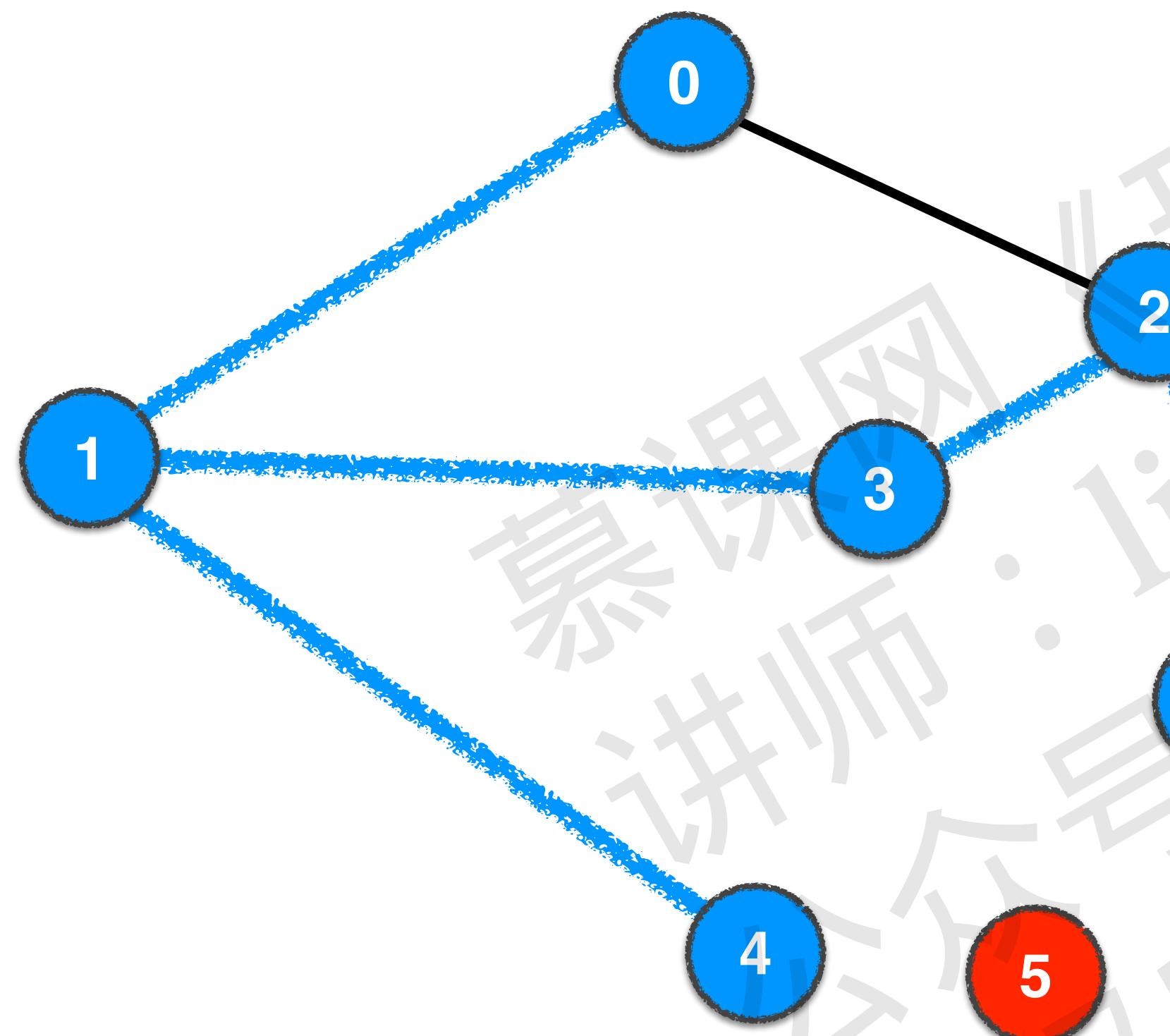
DFS

BFS的重要性质



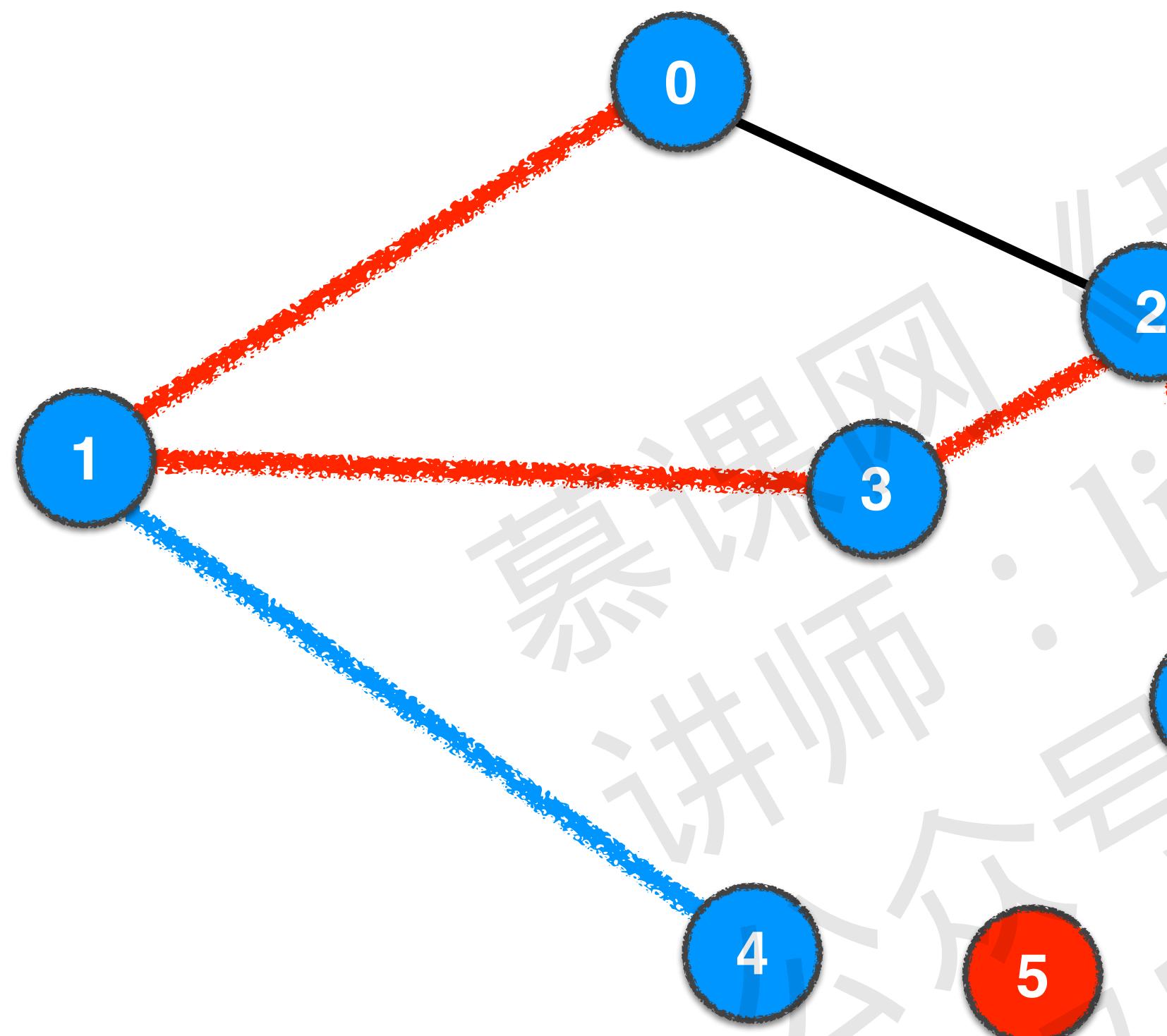
DFS

BFS的重要性质



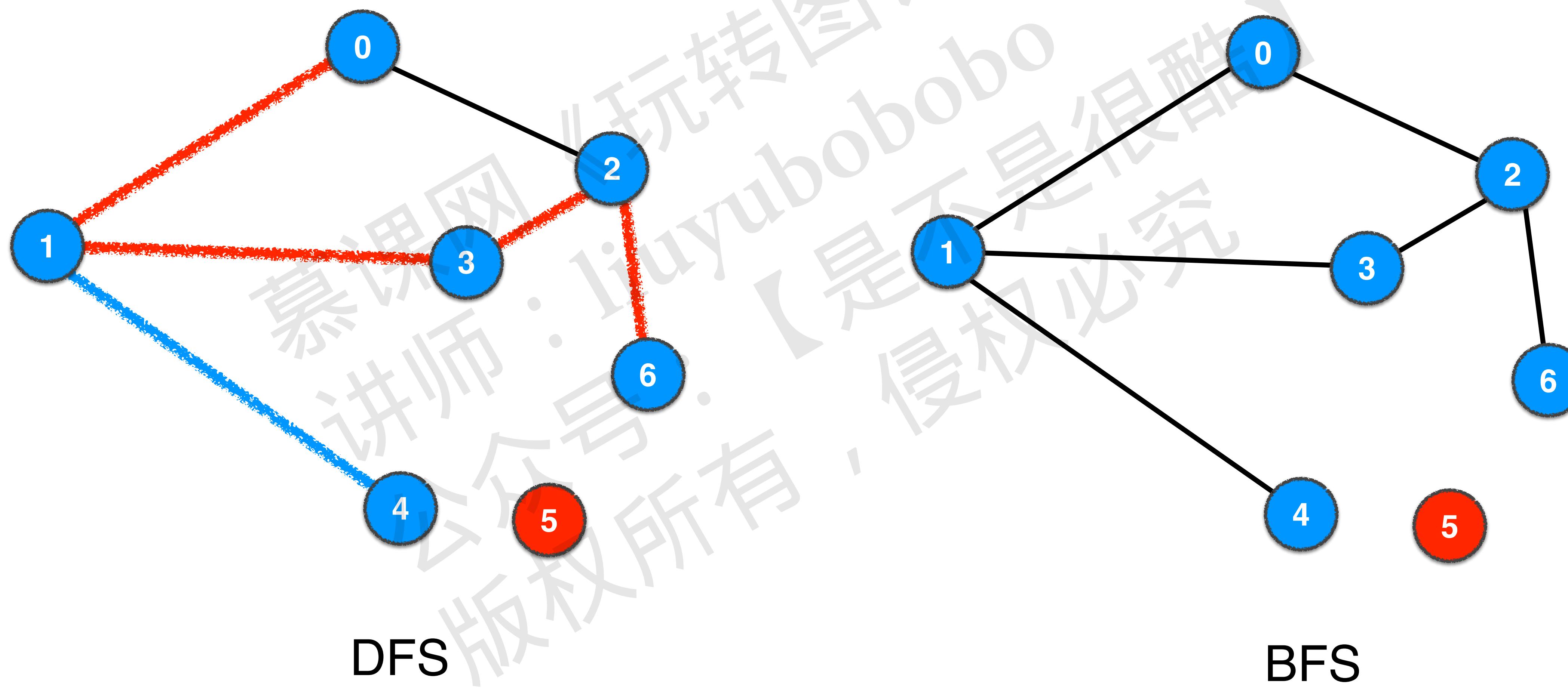
DFS

BFS的重要性质

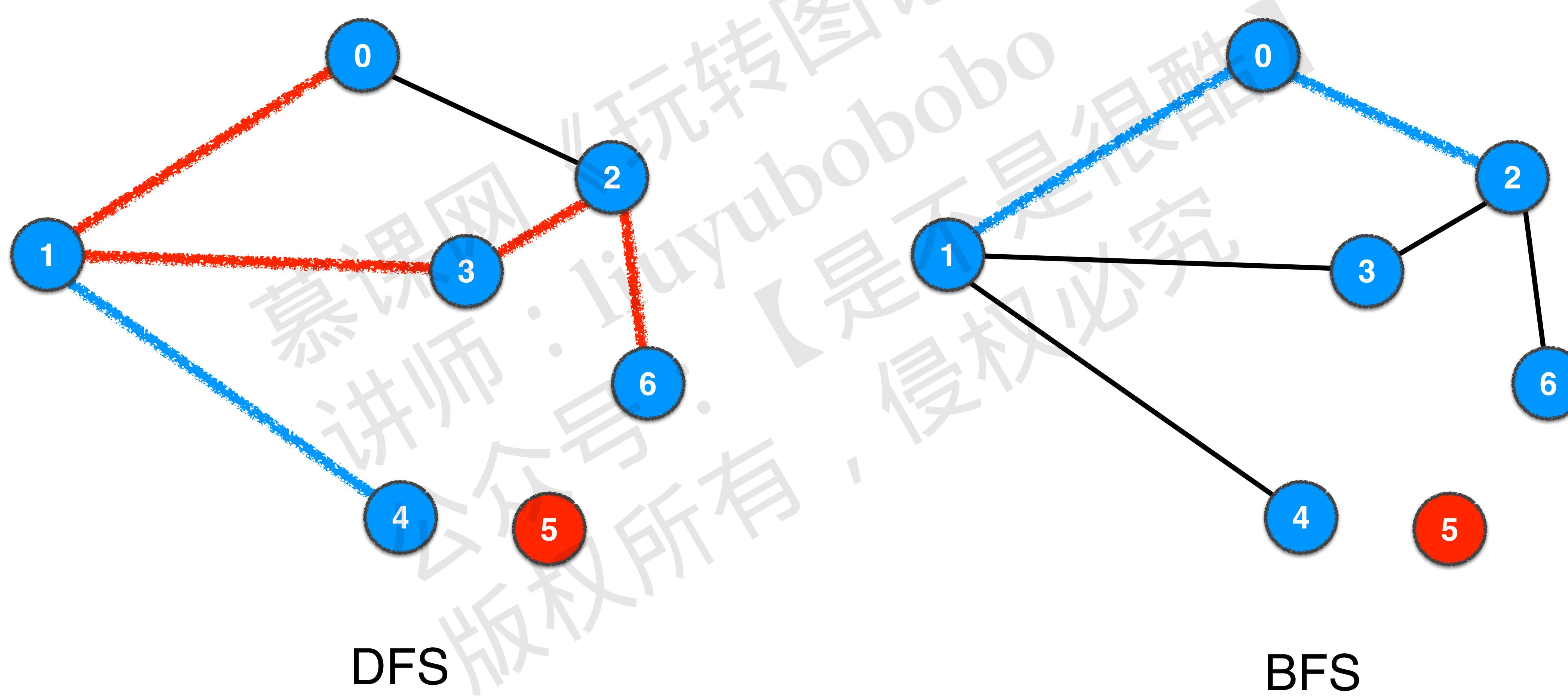


DFS

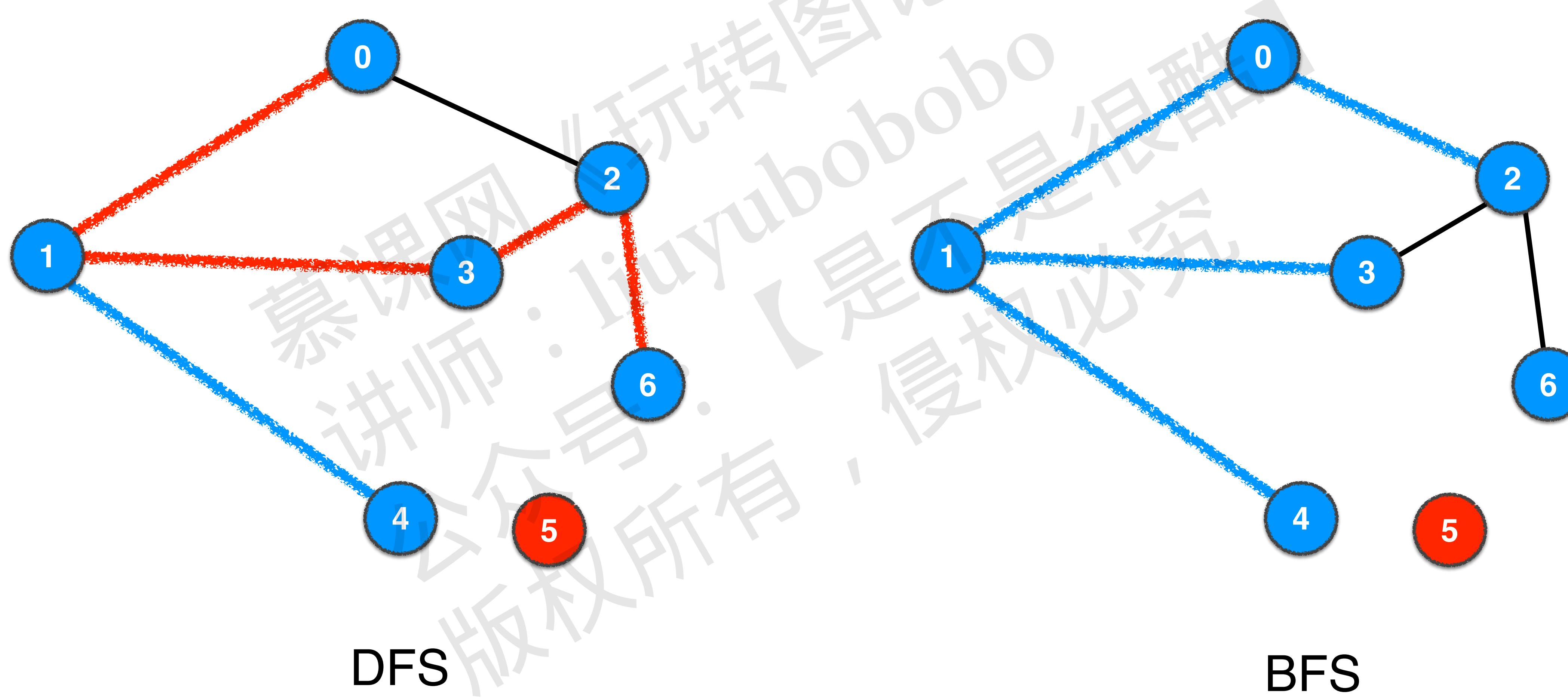
BFS的重要性质



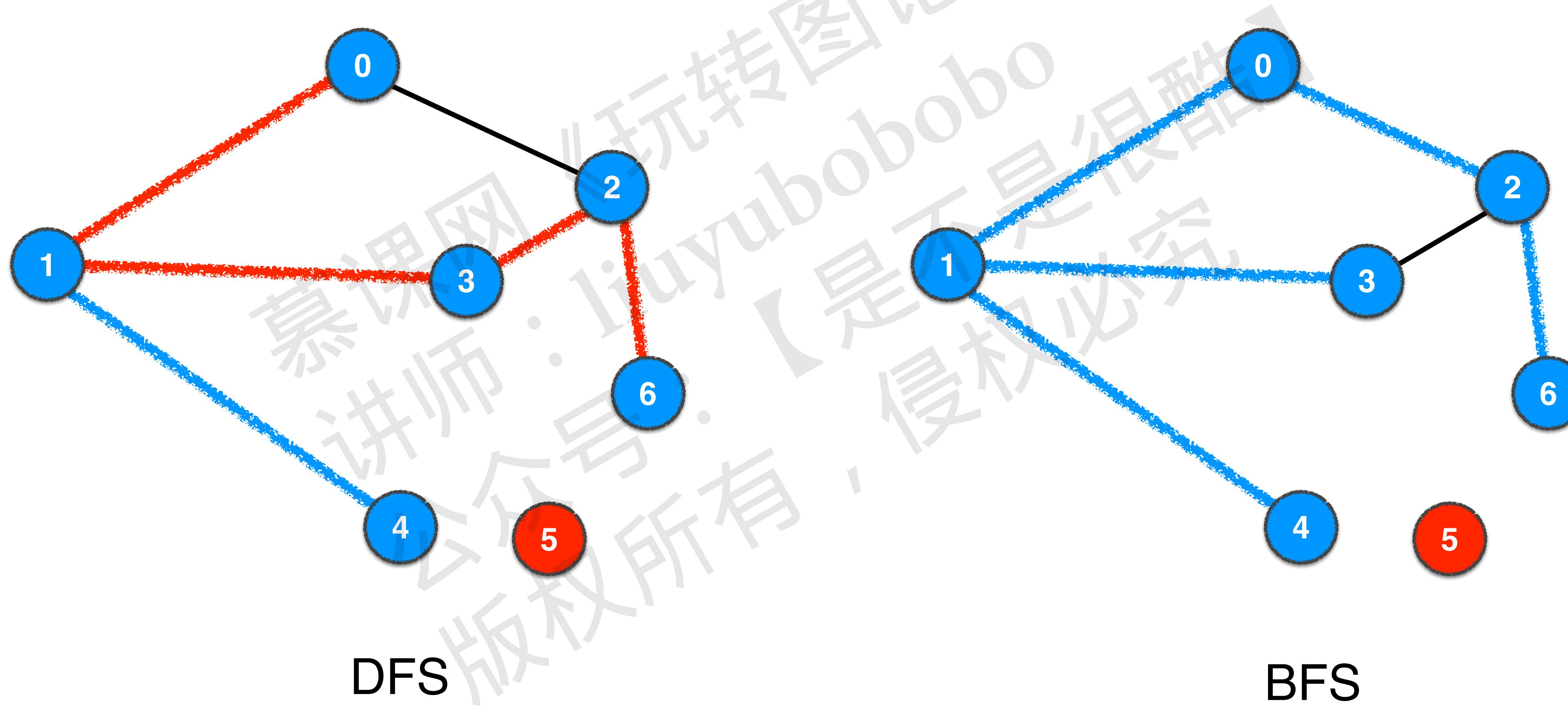
BFS的重要性质



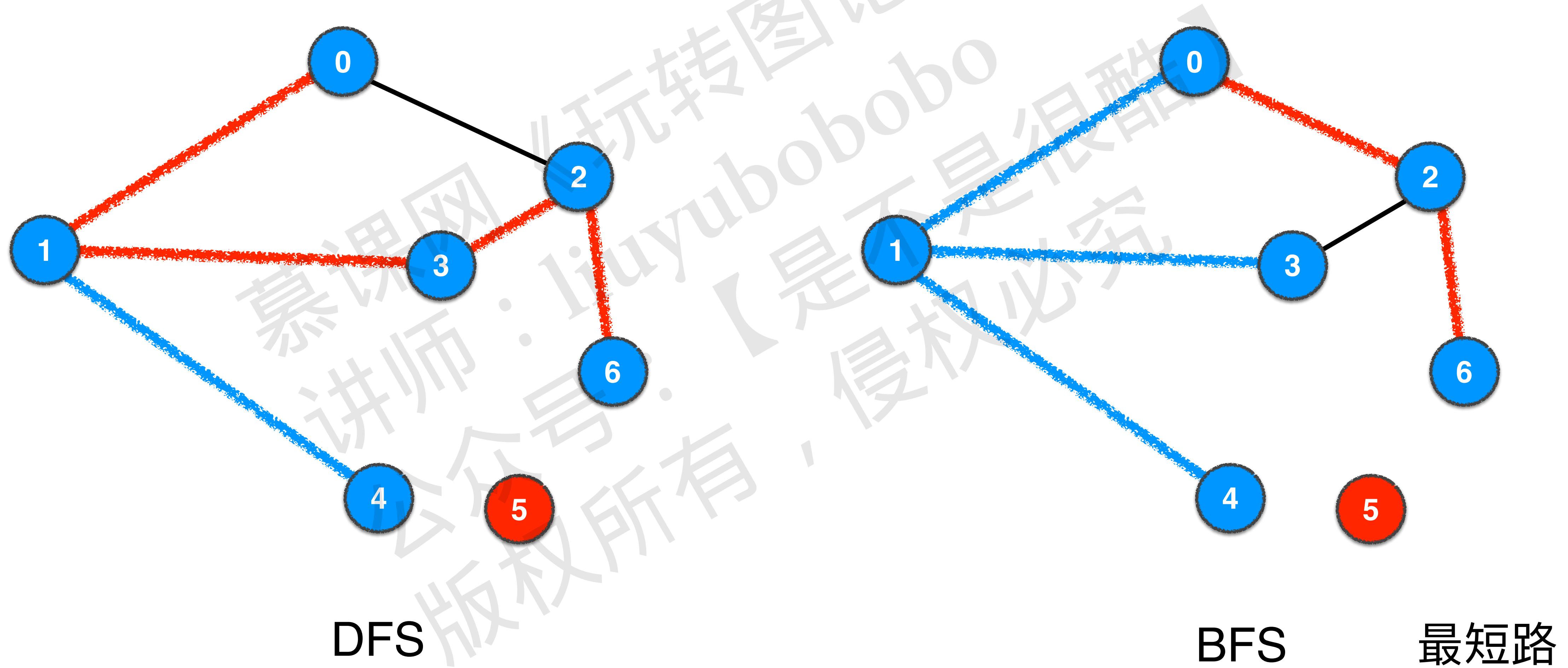
BFS的重要性质



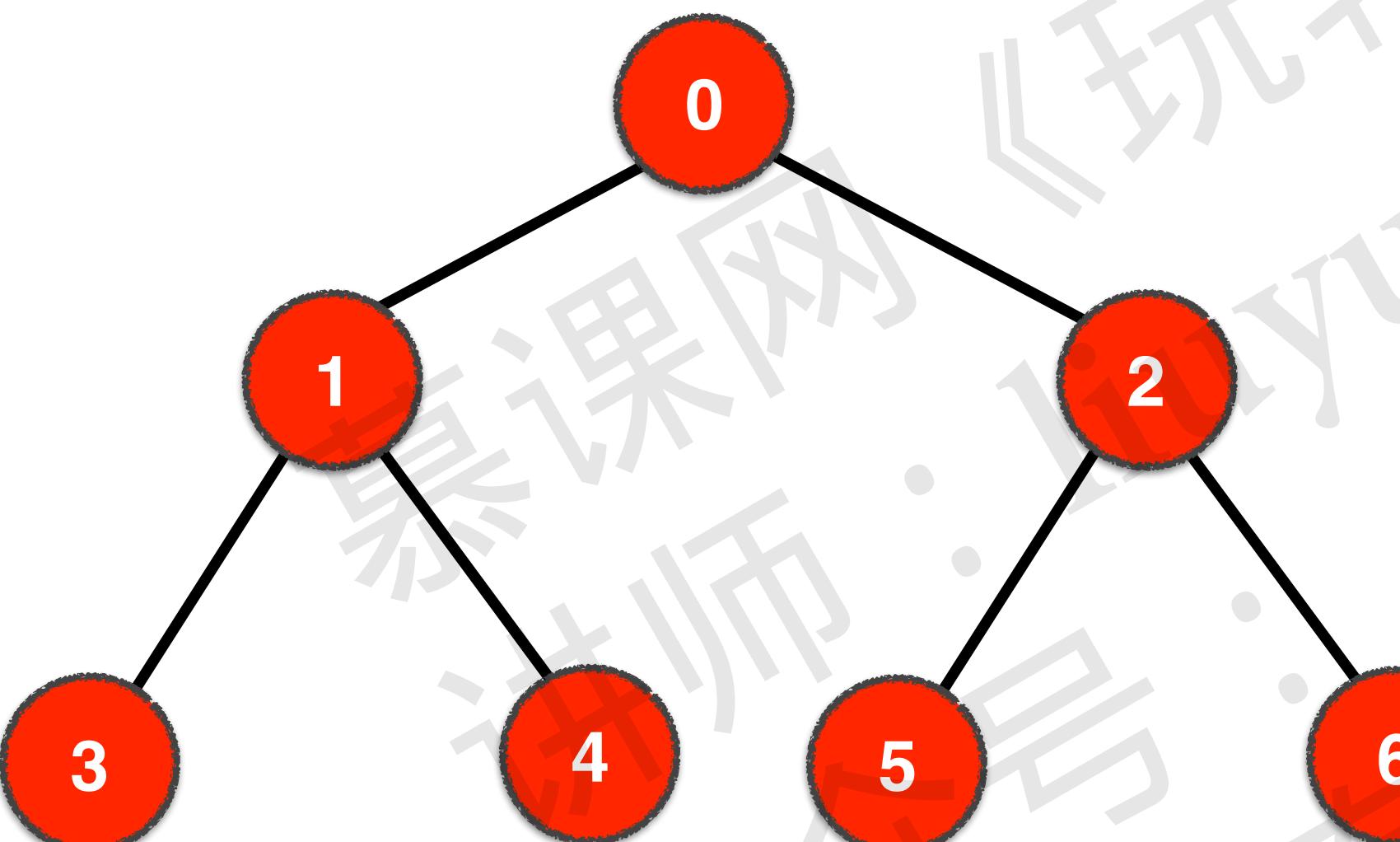
BFS的重要性质



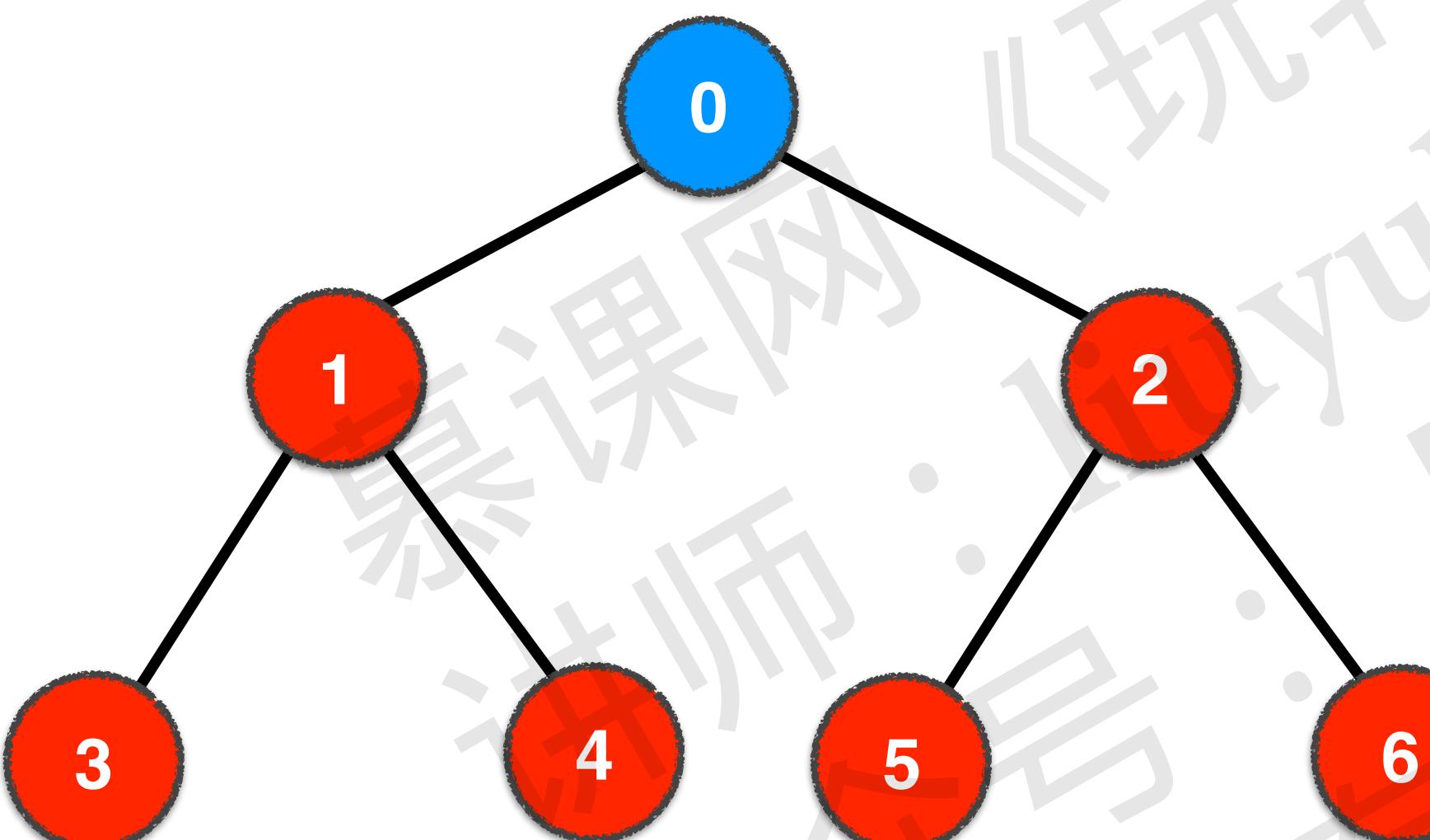
BFS的重要性质



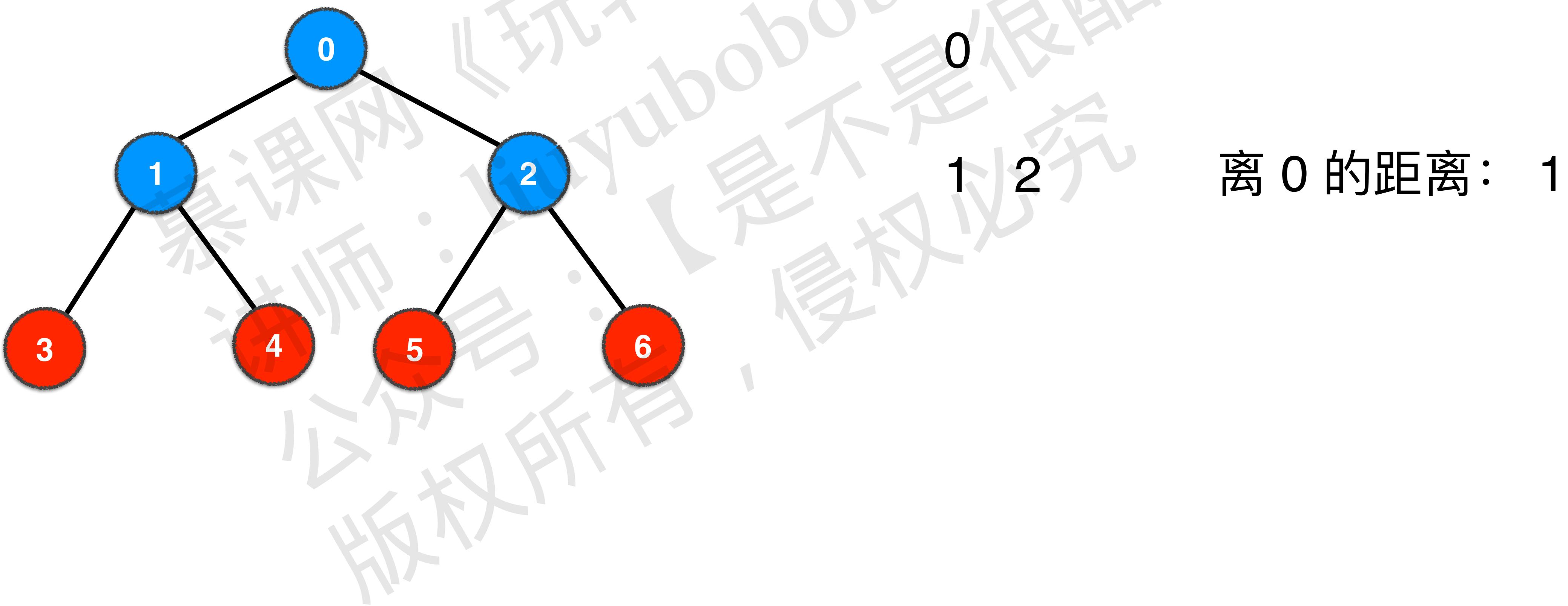
回顾树的广度优先遍历



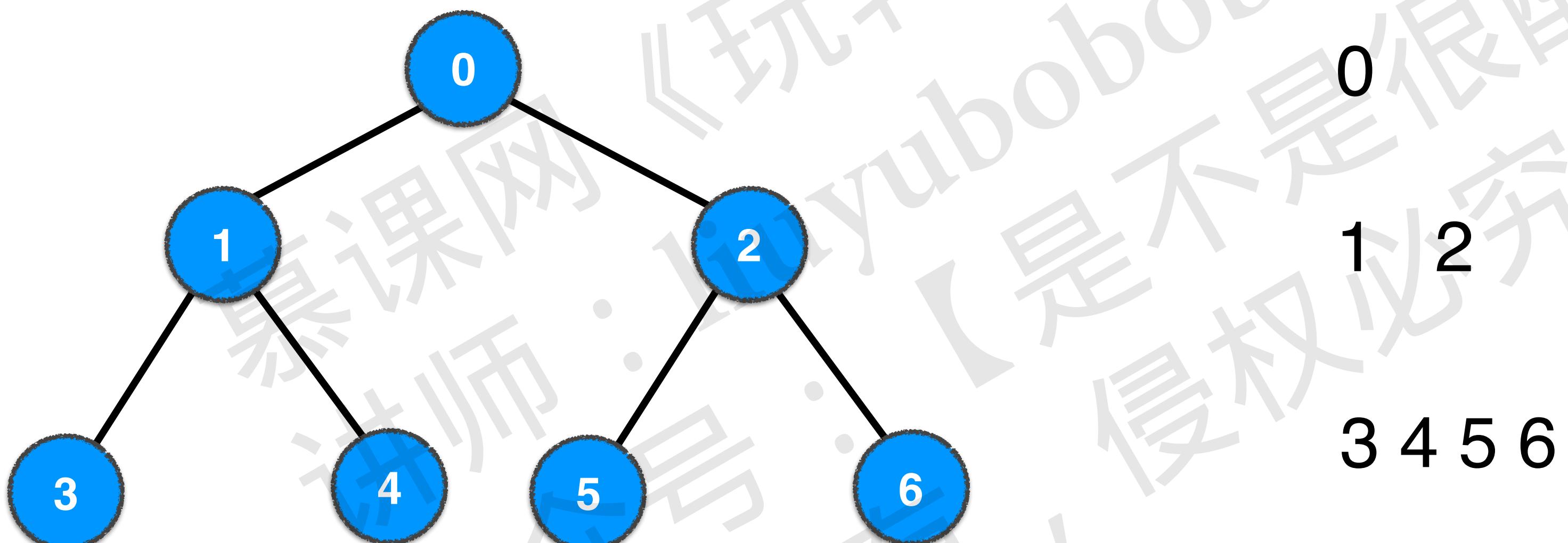
回顾树的广度优先遍历



回顾树的广度优先遍历



回顾树的广度优先遍历



离 0 的距离: 0

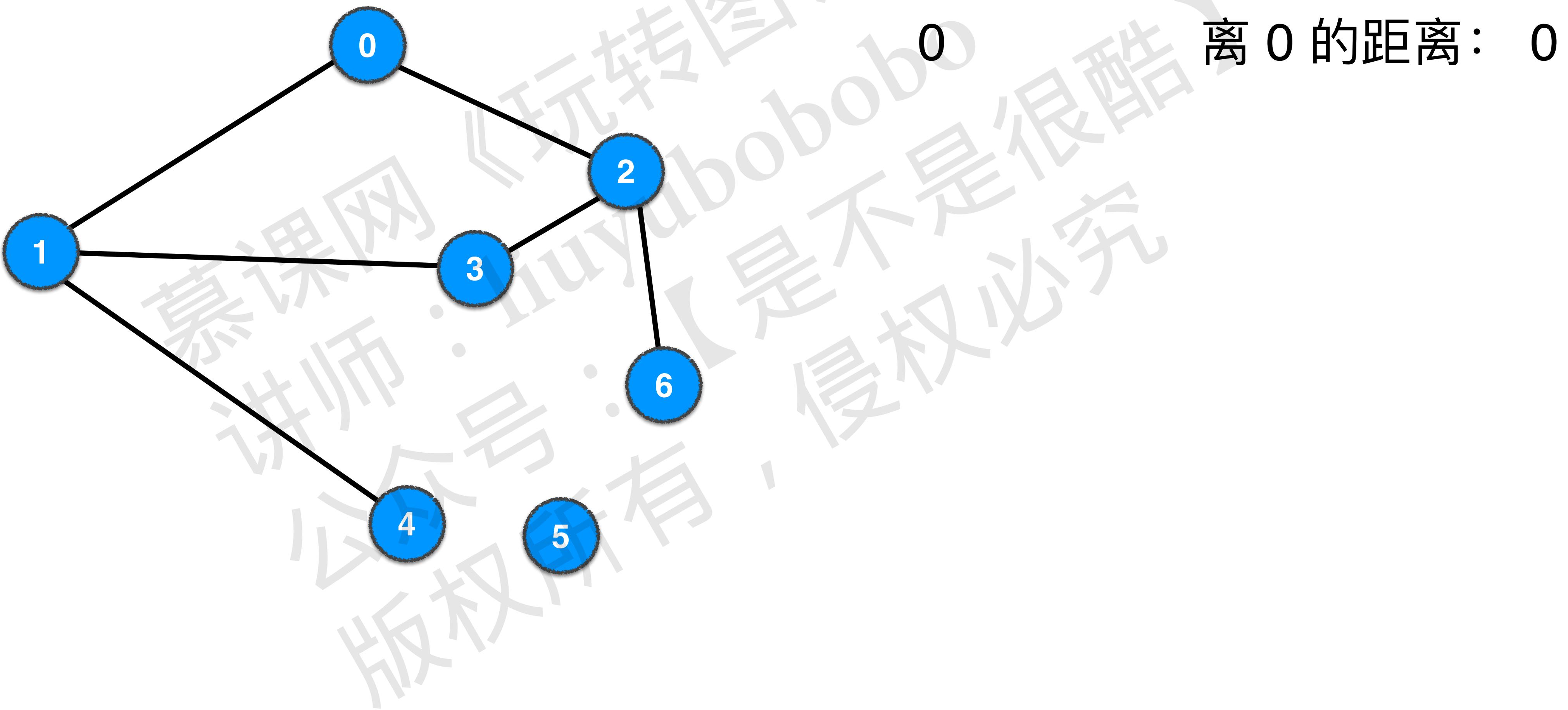
离 0 的距离: 1

离 0 的距离: 2

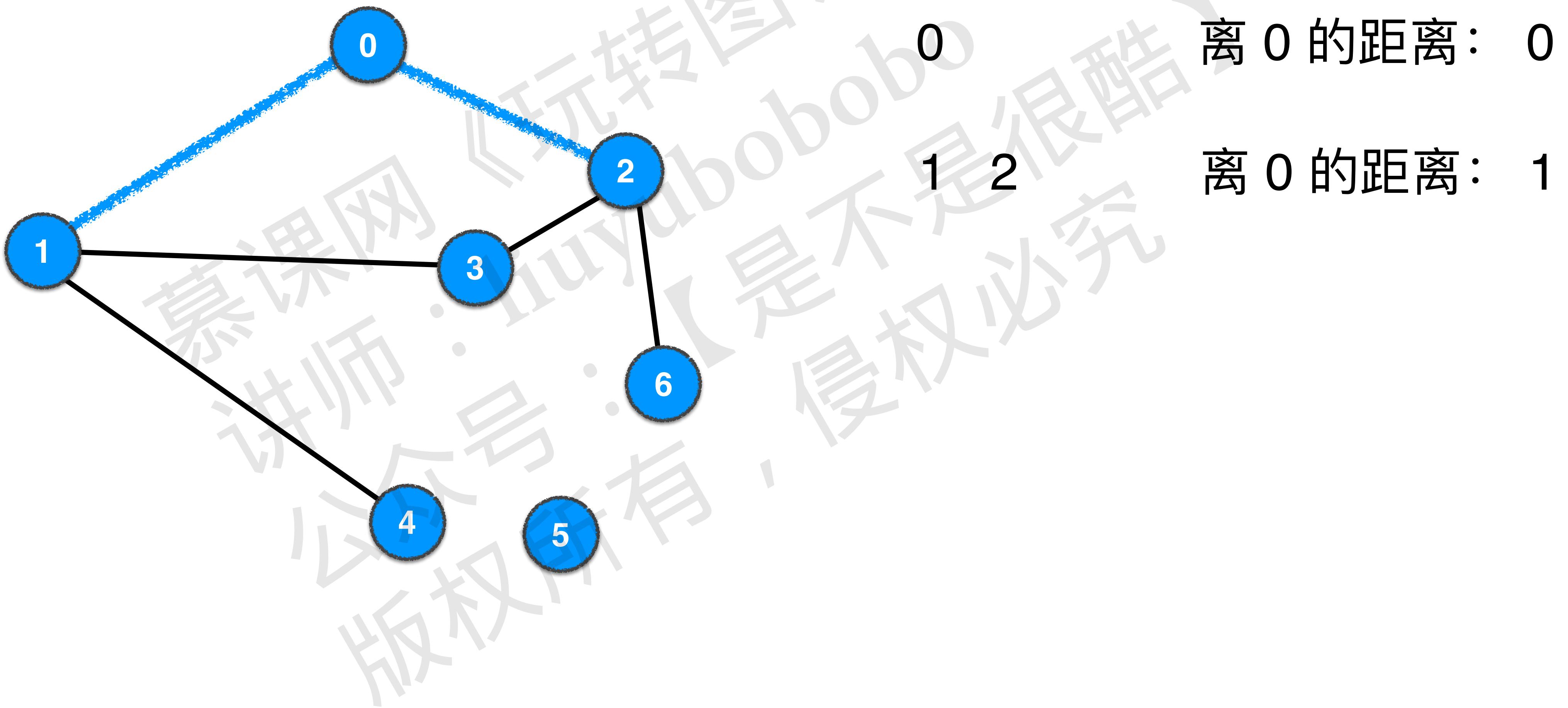
1
2

3
4
5
6

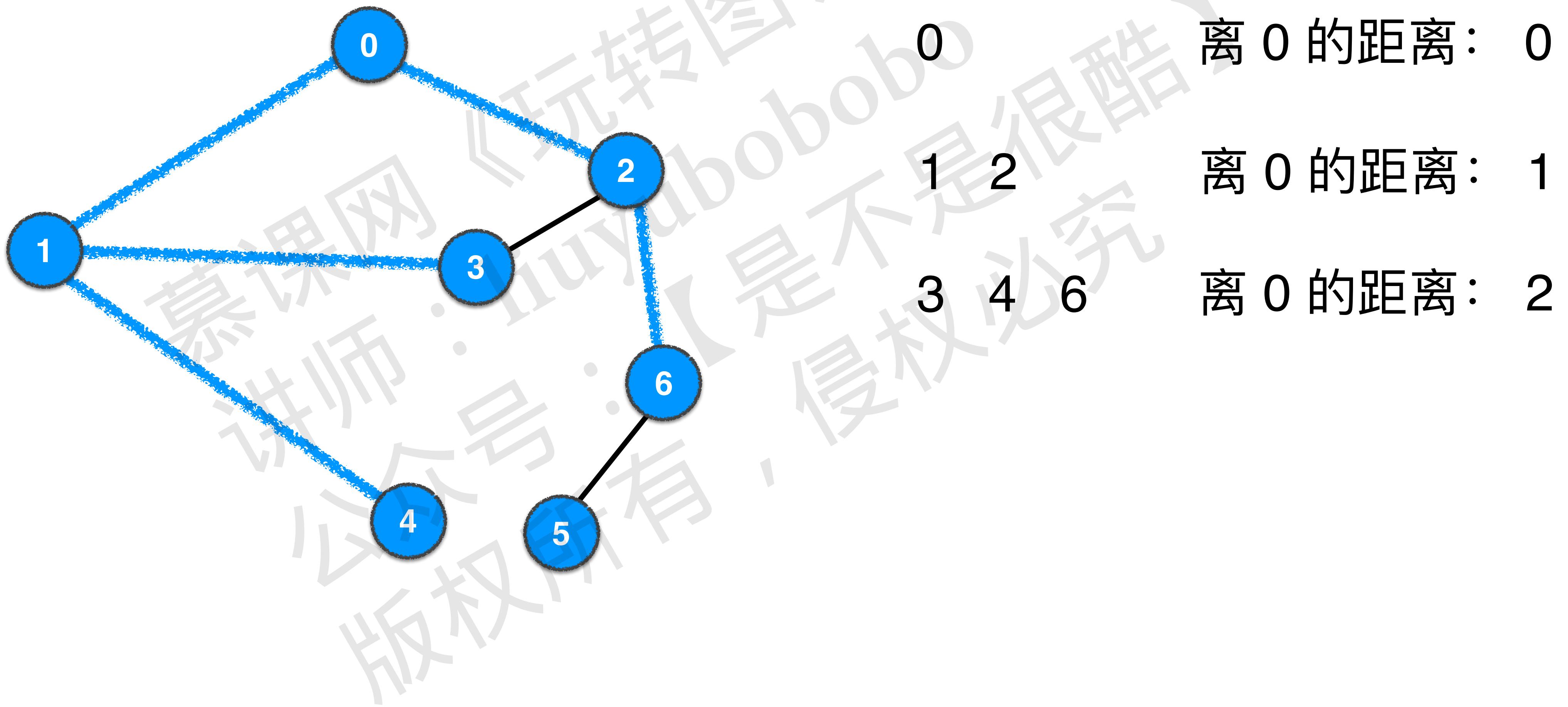
BFS的重要性质



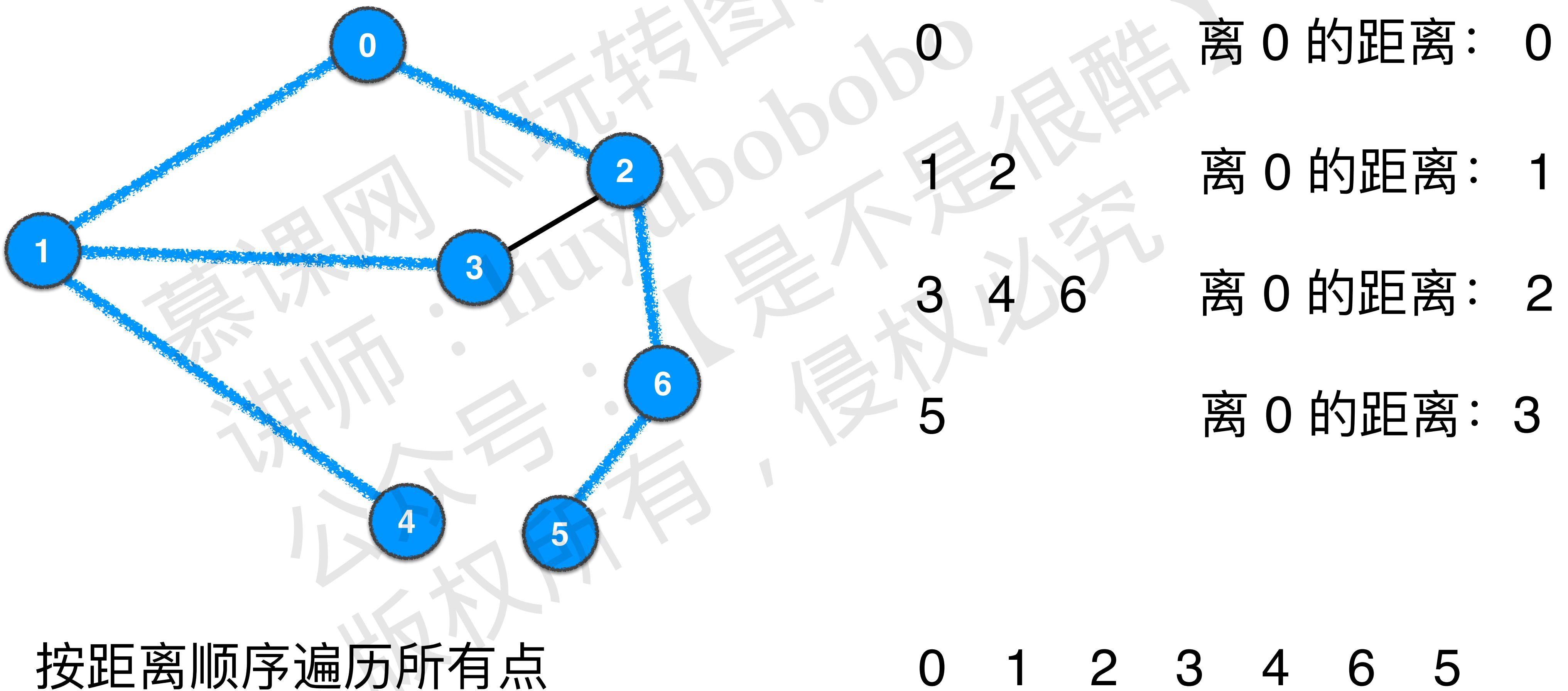
BFS的重要性质



BFS的重要性质



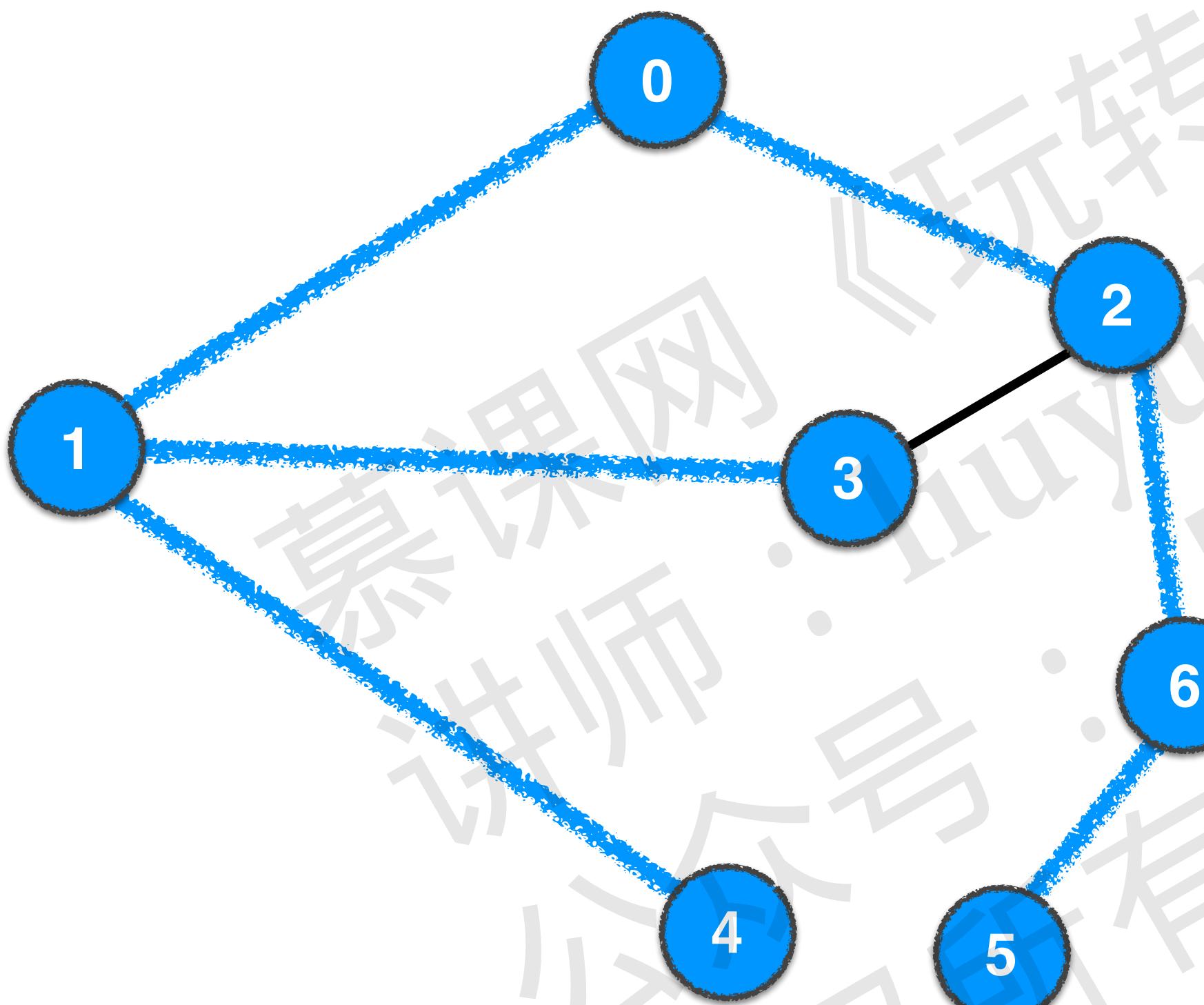
BFS的重要性质



无权图的最短路径

liuyubobobo

BFS的重要性质



按距离顺序遍历所有点

离 0 的距离:	0	1	2	3	4	5	6
离 0 的距离: 0	0						
离 0 的距离: 1		1					
离 0 的距离: 2			2				
离 0 的距离: 3				3			
dis	0	1	1	2	2	2	3
	0	1	2	3	4	6	5

编程实践：使用BFS解决无权图最短路问题

慕课网《玩转图论算法》
讲师：liuyibobobo
版权所有，侵权必究

无权图最短路径问题

单源最短路径问题

所有点对最短路径问题

提前返回?

比较 BFS 和 DFS，本章小结

liuyubobobo

本章小结

图的广度优先遍历 BFS

无权图的最短路径

无权图，无权图，无权图！

比较 BFS 和 DFS

```
visited[0...V-1] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v);  
  
dfs(int s){  
  
    stack.add(s)  
    visited[s] = true;  
    while(!q.isEmpty()){  
        int v = stack.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                stack.add(w);  
                visited[w] = true;  
            }  
    }  
}
```

```
visited[0...V-1] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        bfs(v);  
  
bfs(int s){  
  
    queue.add(s)  
    visited[s] = true;  
    while(!q.isEmpty()){  
        int v = queue.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                queue.add(w);  
                visited[w] = true;  
            }  
    }  
}
```

比较 BFS 和 DFS

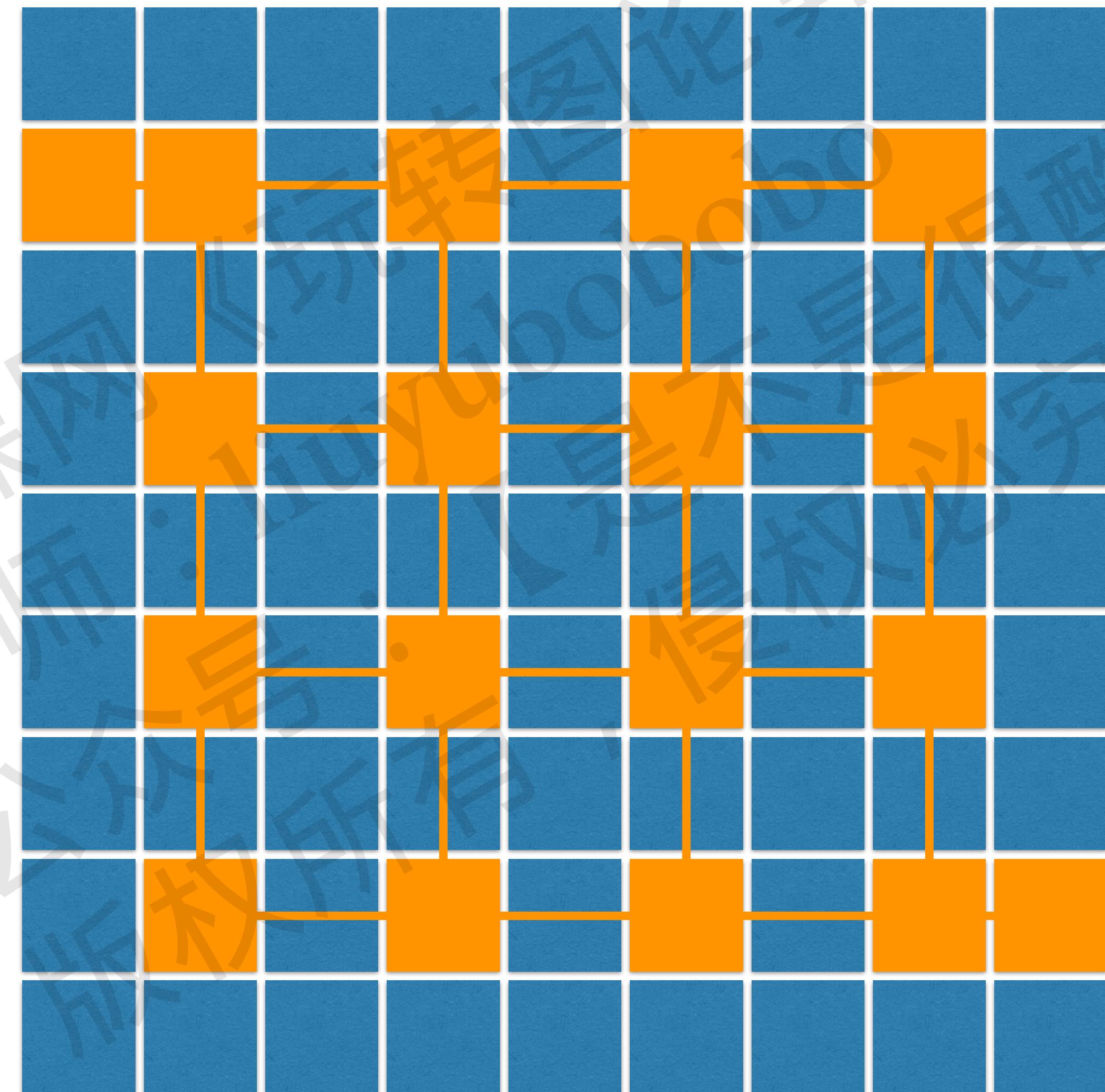
```
visited[0...V-1] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        dfs(v);  
  
dfs(int s){  
  
    stack.add(s)  
    visited[s] = true;  
    while(!stack.isEmpty()){  
        int v = stack.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                stack.add(w);  
                visited[w] = true;  
            }  
    }  
}
```

```
visited[0...V-1] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        bfs(v);  
  
bfs(int s){  
  
    queue.add(s)  
    visited[s] = true;  
    while(!queue.isEmpty()){  
        int v = queue.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                queue.add(w);  
                visited[w] = true;  
            }  
    }  
}
```

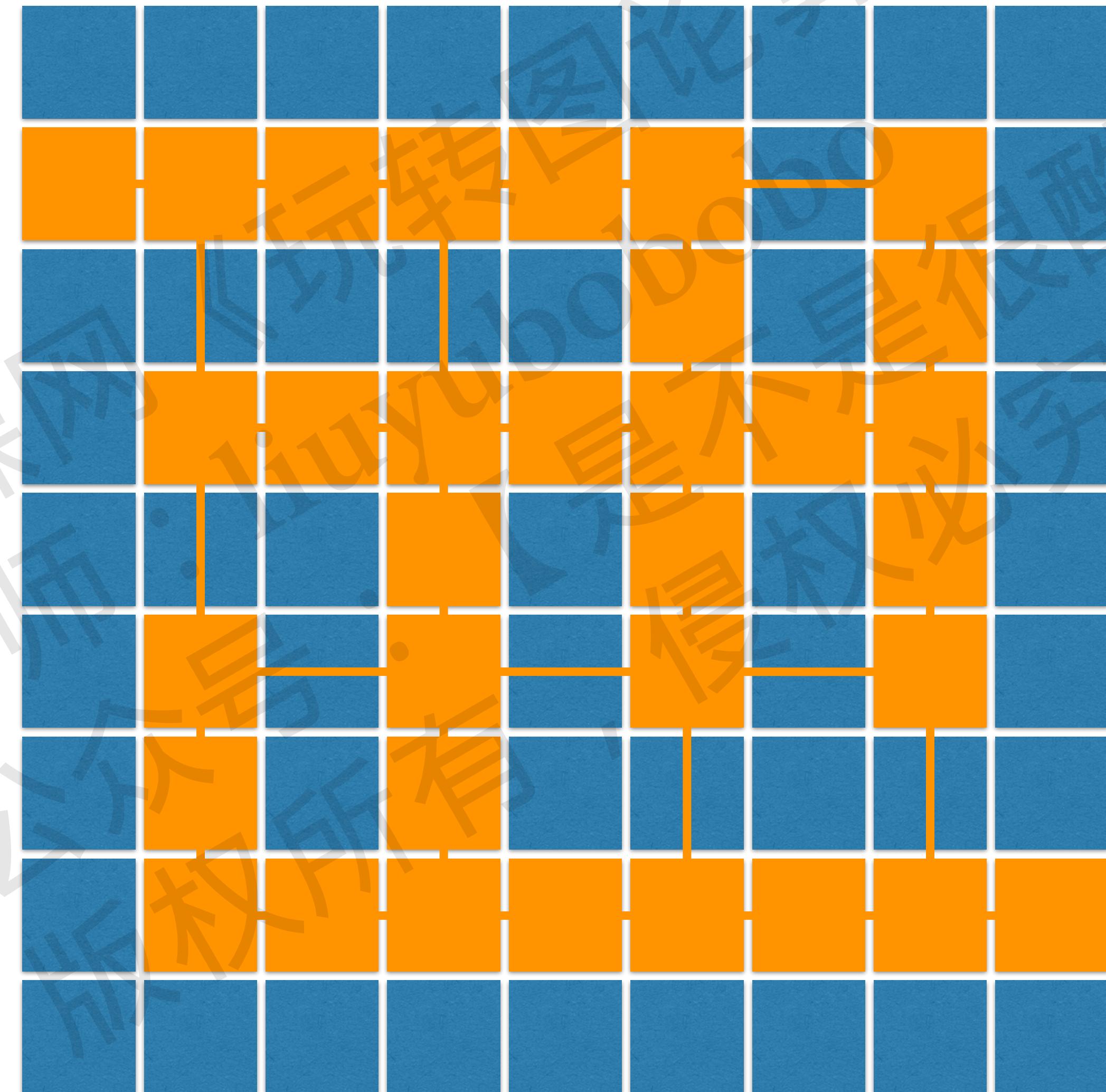
比较 BFS 和 DFS

```
visited[0...V-1] = false;  
  
for(int v = 0; v < V; v ++)  
    if(!visited[v])  
        search(v);  
  
search(int s){  
    x.add(s)  
    visited[s] = true;  
    while(!x.isEmpty()){  
        int v = x.remove();  
  
        for(int w: G.adj(v))  
            if(!visited[w]){  
                x.add(w);  
                visited[w] = true;  
            }  
    }  
}
```

如何生成迷宫



如何生成迷宫



如何生成迷宫

视频演示：迷宫生成

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

看得见的算法

The screenshot shows a course landing page with the following details:

- Category:** / 实战 / 看得见的算法 7个经典应用诠释算法精髓
- Title:** 深度实战玩转算法
- Description:** 以Java语言主讲，通过7款经典好玩游戏，真正将算法用于实际开发
- Price:** ¥ 248.00
- Payment Options:** 花呗付款, 京东白条
- Difficulty:** 难度 中级
- Duration:** 时长 17小时
- Students:** 学习人数 1178
- Rating:** 好评度 100%
- Buttons:** 收藏 (Collection), 分享 (Share), 进入课程 (Enter Course)

本章小结

这个课程主要基于面试

下一章：看Leetcode上的图论问题

图论建模

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