

玩儿转图论算法

liuyubobo

最短路径

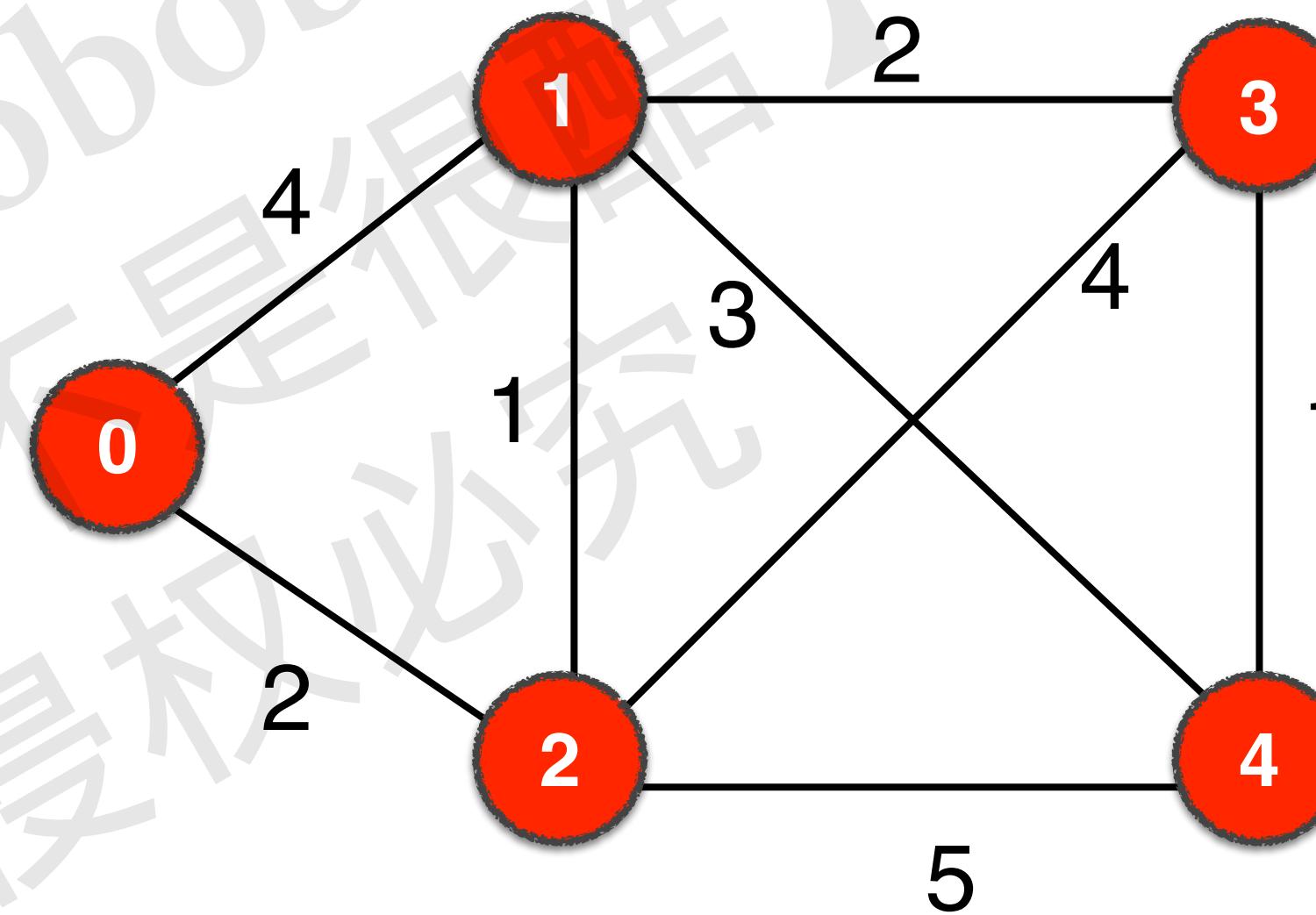
liuyubobobo

Dijkstra 和单源最短路径

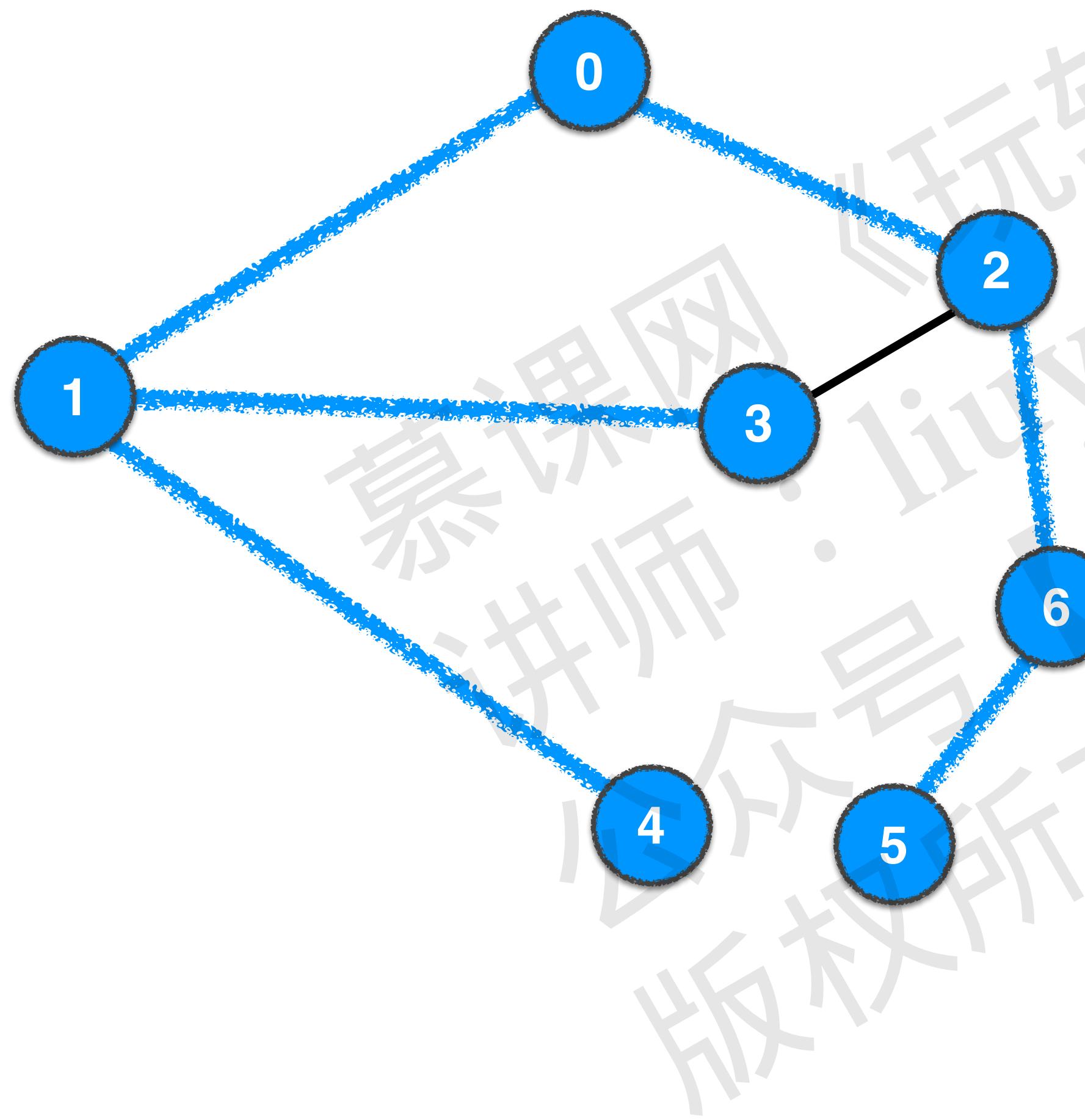
liuyubobobo

最短路径

路径规划



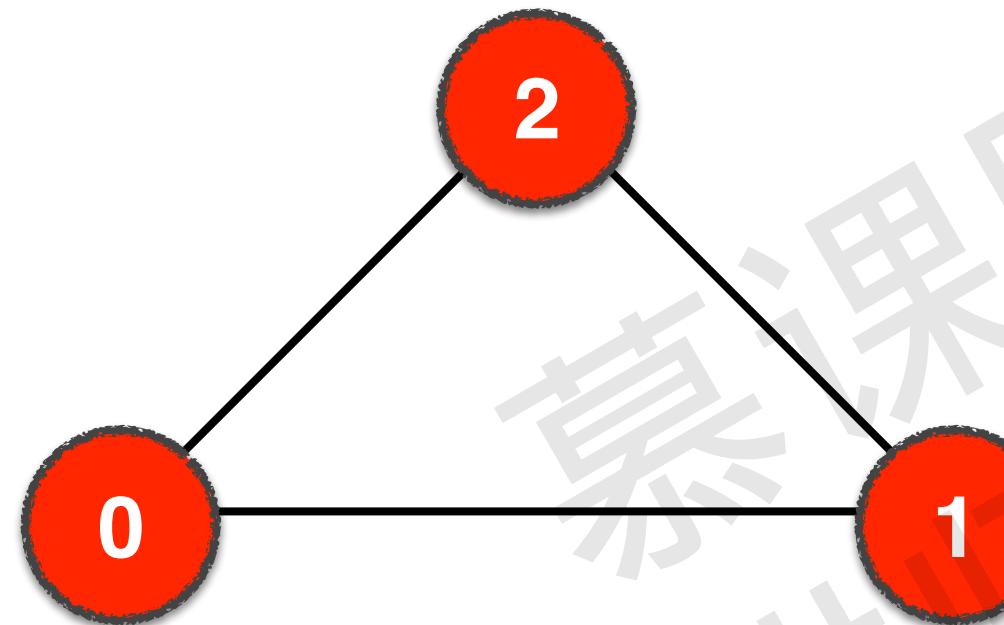
单源最短路径



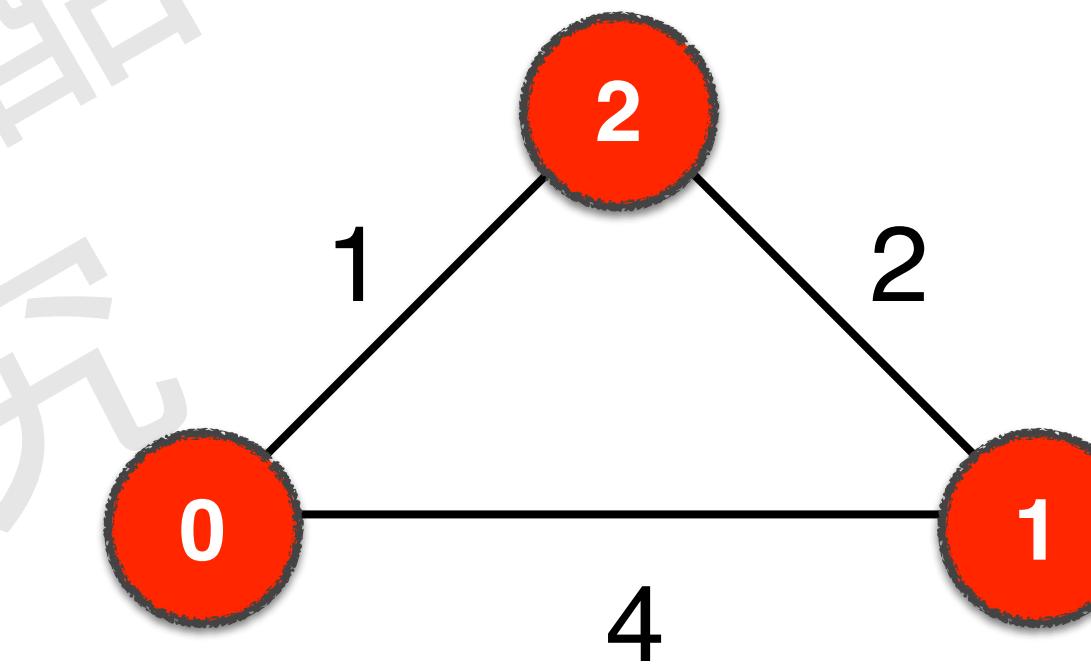
dis	0	1	2	3	4	5	6
0	1	1	2	2	3	3	2

求解出了从源点 s 到图中任意点的
最短路径

带权图的最短路径



对于无权图，
0-1 的最短路径肯定直接走 0-1 边



对于带权图，
0-1 的最短路径是绕道走2

带权图的最短路径

Dijkstra 算法

不能处理负权边

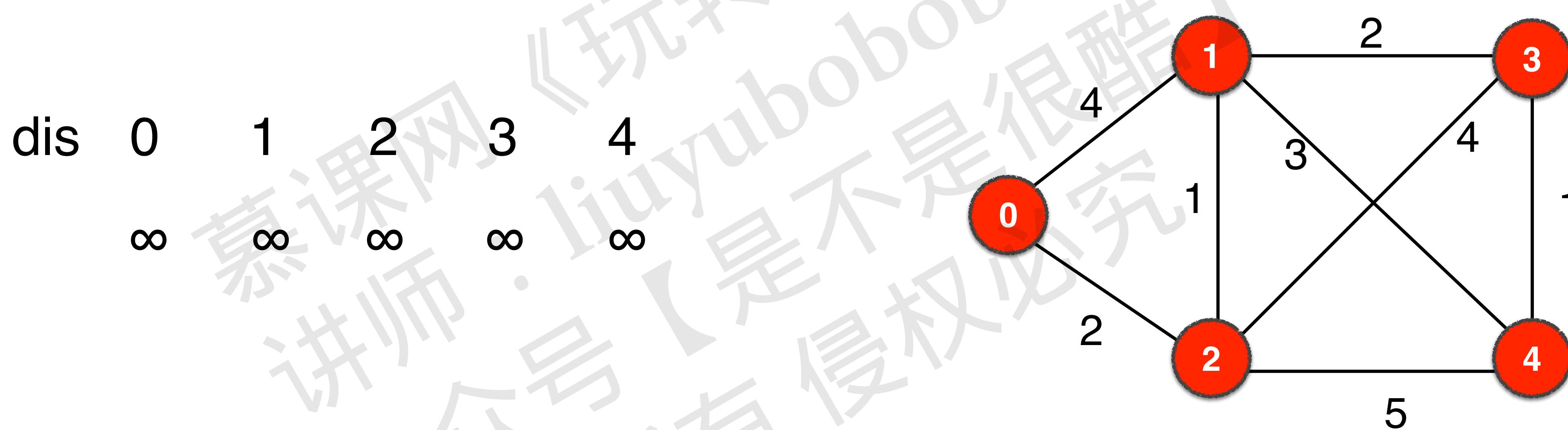
大量应用不依赖负权边

处理负权边会使得复杂度上升

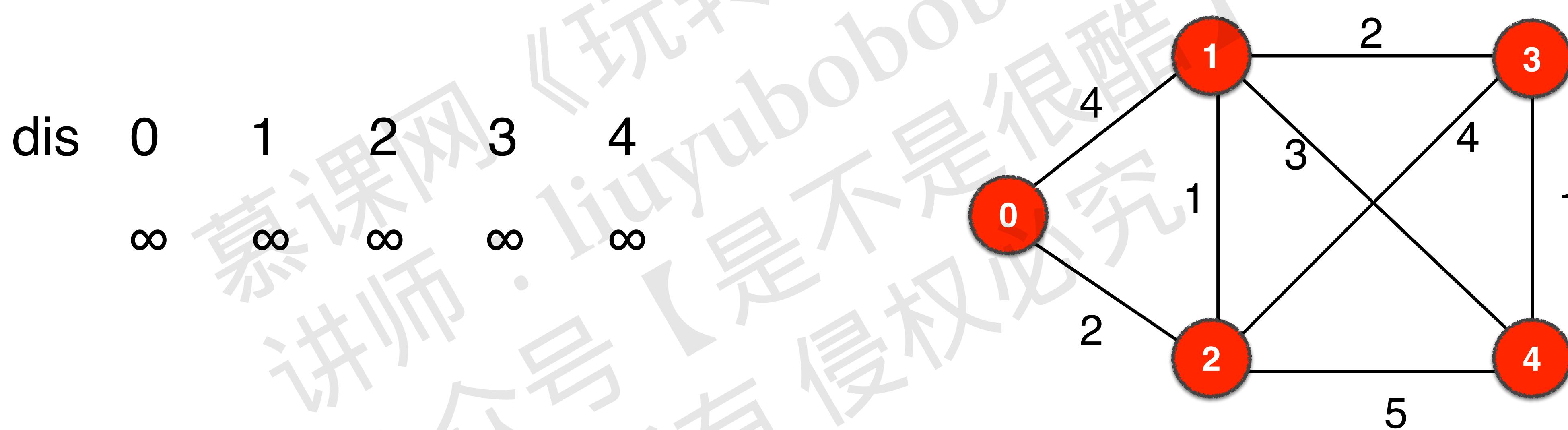
Dijkstra 算法原理及模拟

liuyubobobo

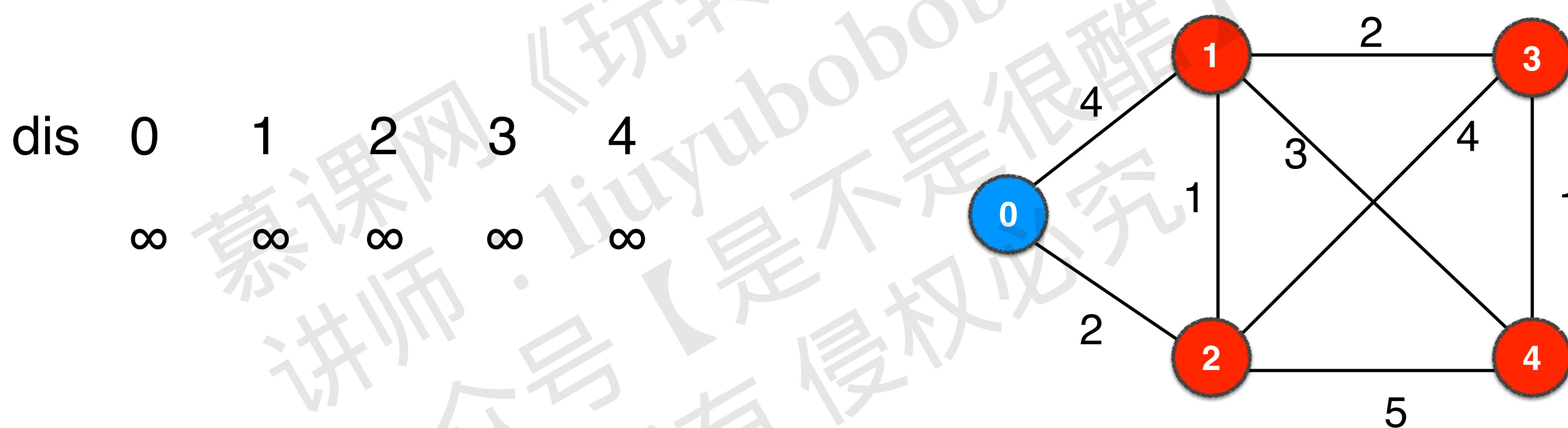
Dijkstra 算法



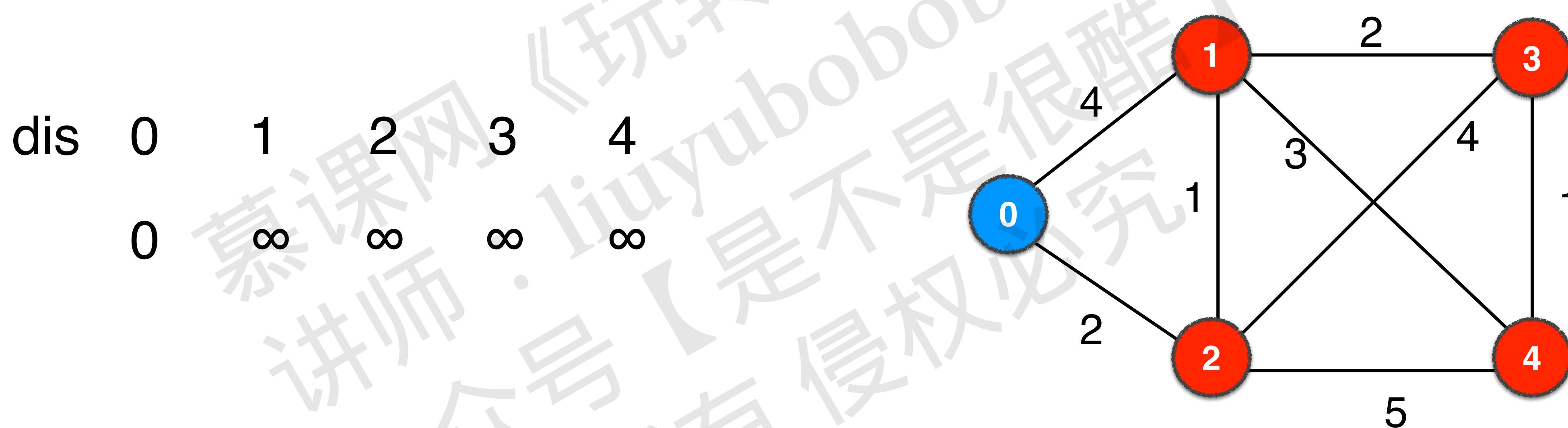
Dijkstra 算法



Dijkstra 算法



Dijkstra 算法



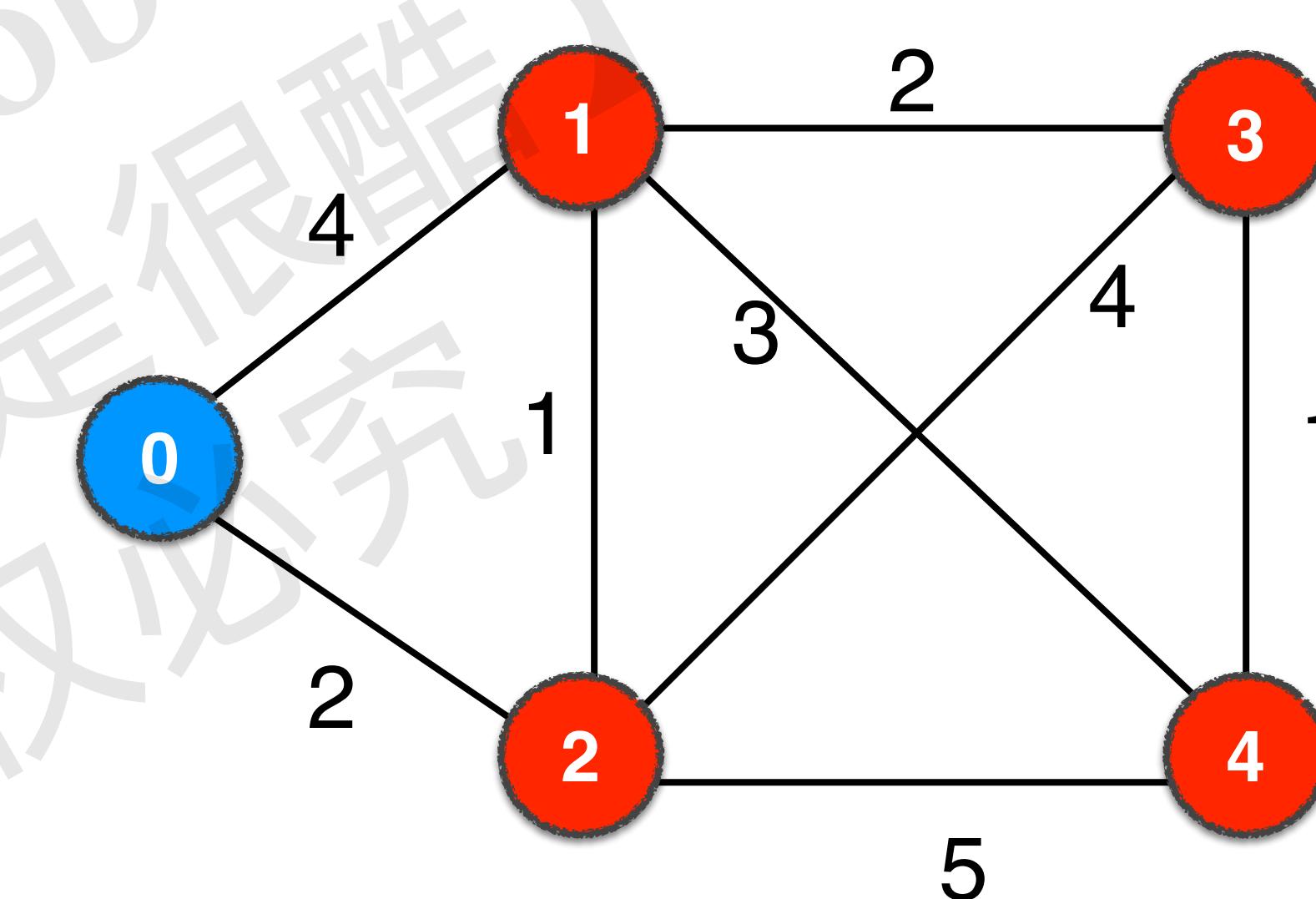
Dijkstra 算法

dis

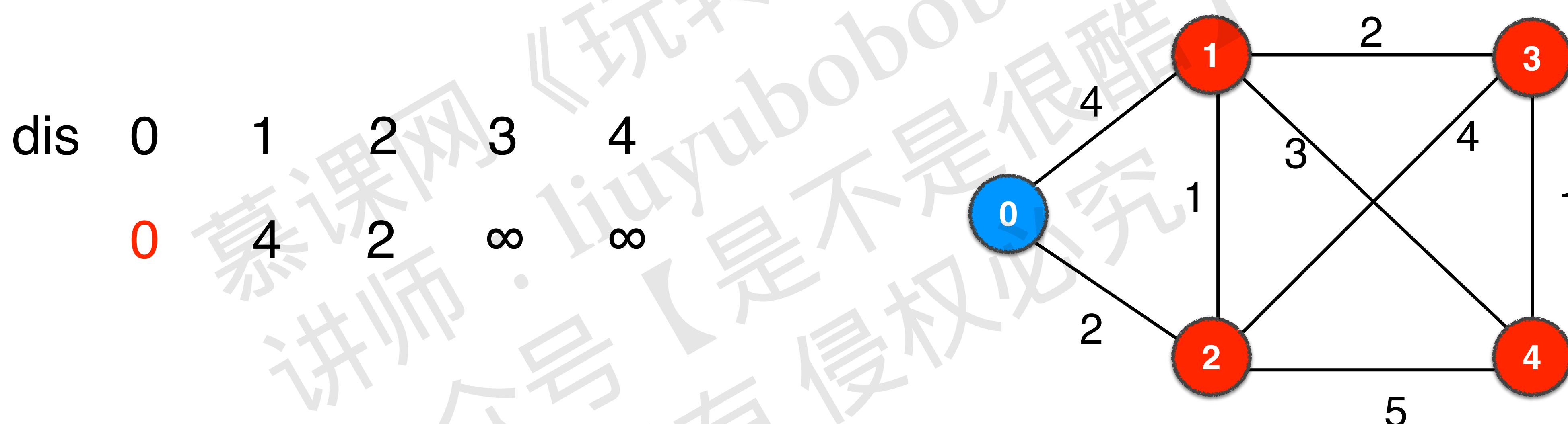
1

3

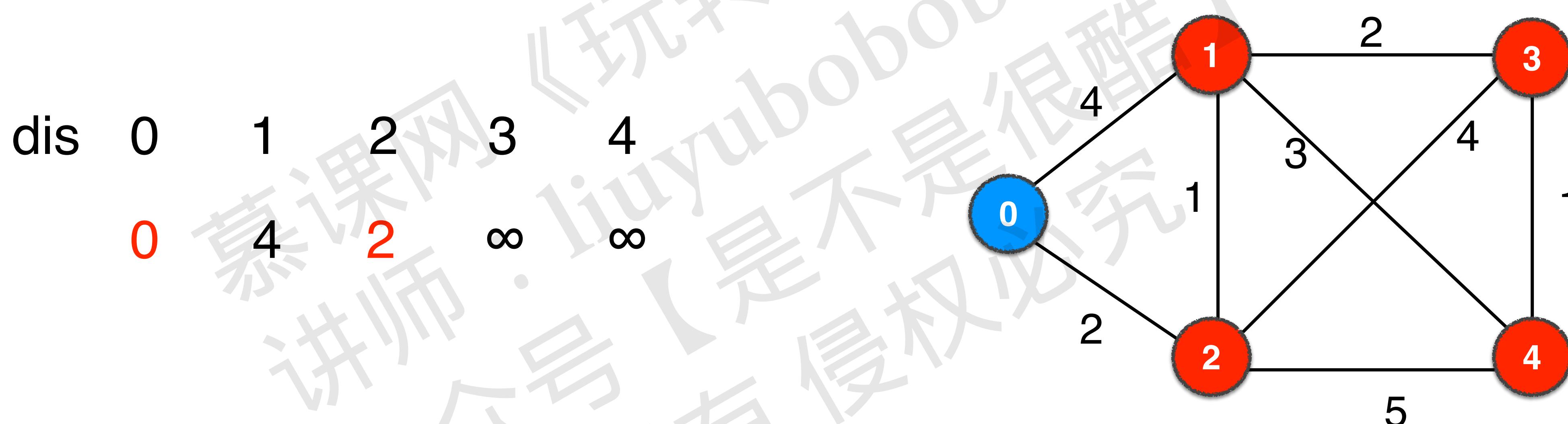
4



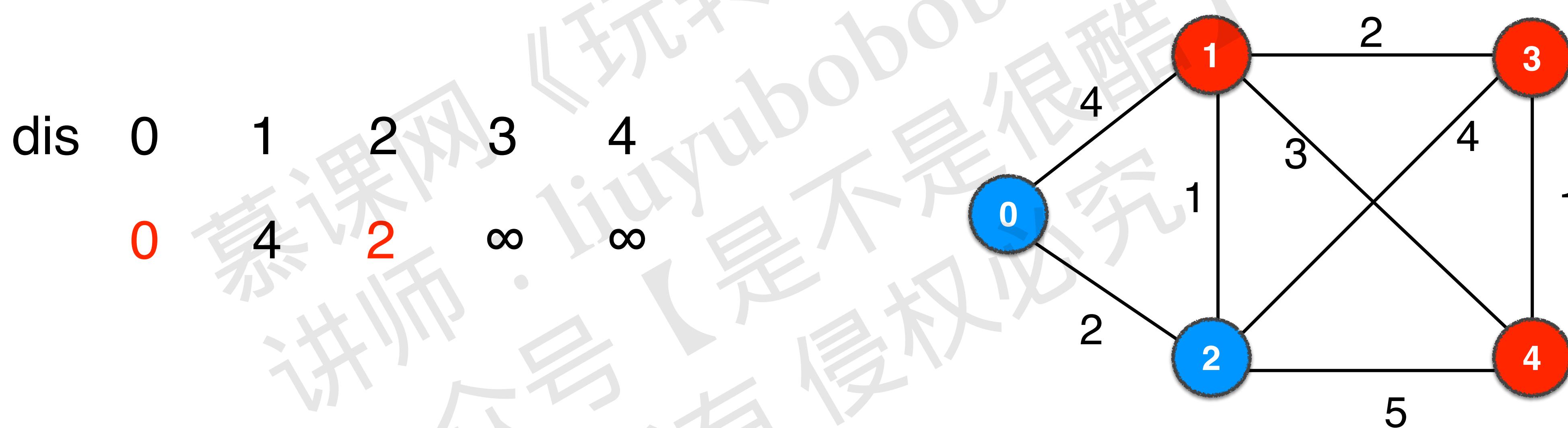
Dijkstra 算法



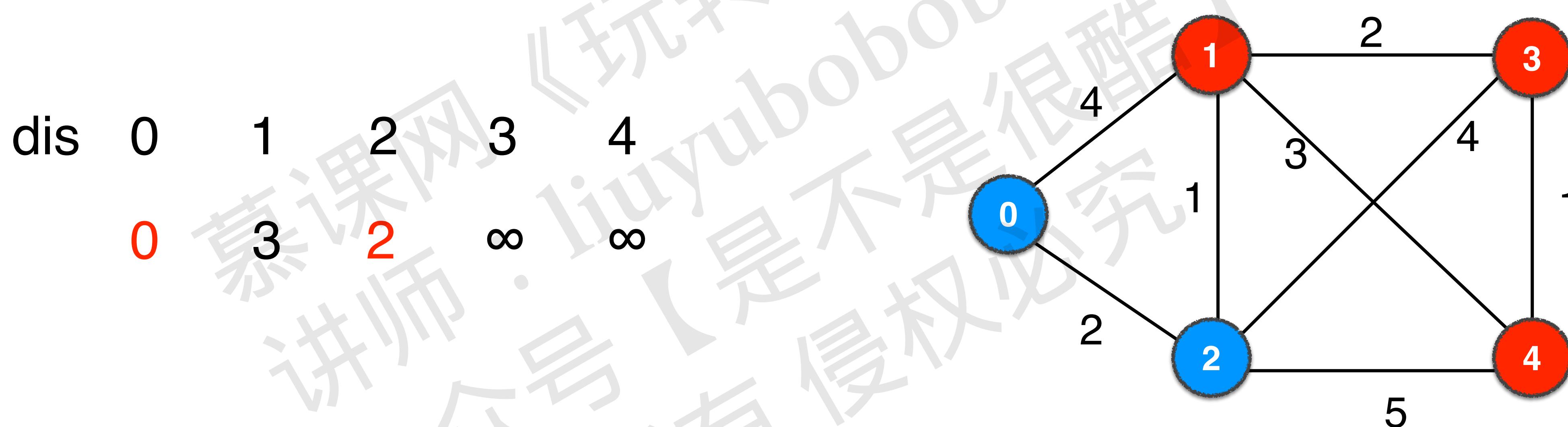
Dijkstra 算法



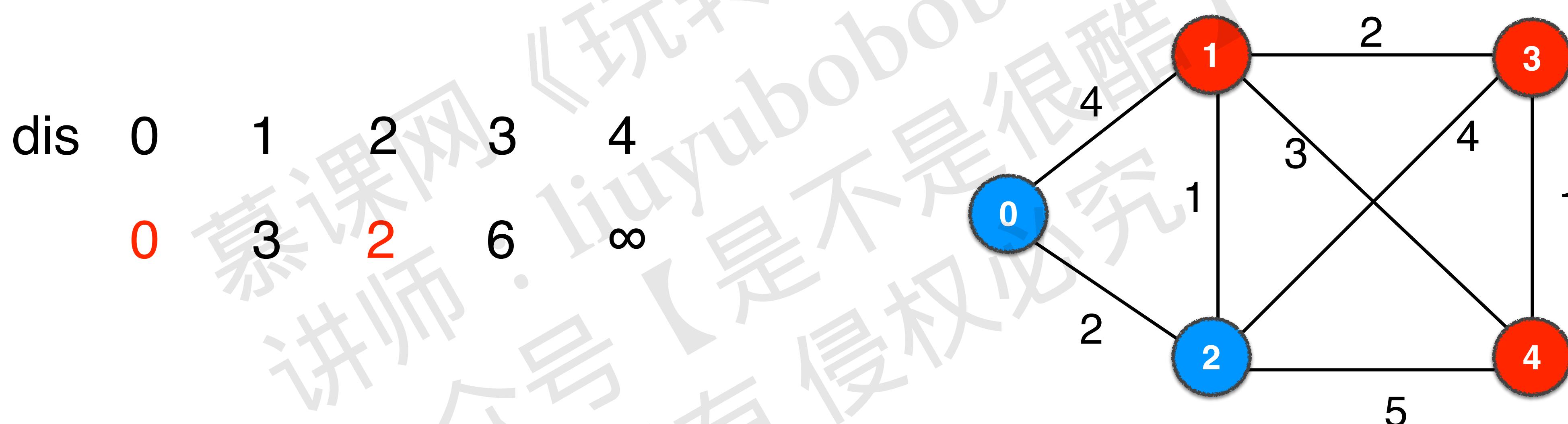
Dijkstra 算法



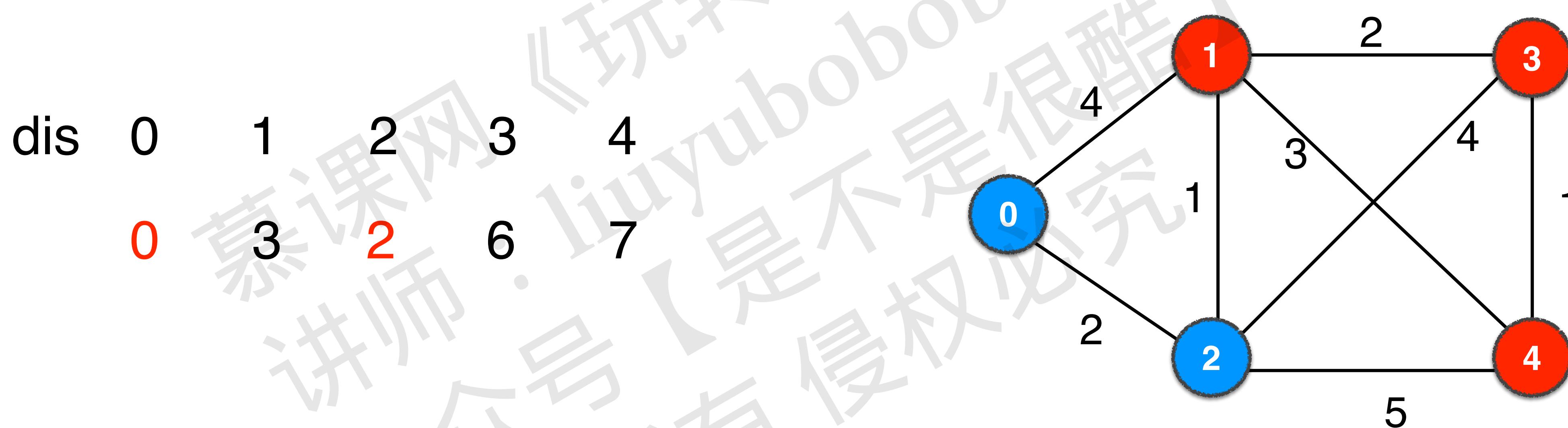
Dijkstra 算法



Dijkstra 算法



Dijkstra 算法



Dijkstra 算法

dis

2
2

3
6

11

0

卷之三

1

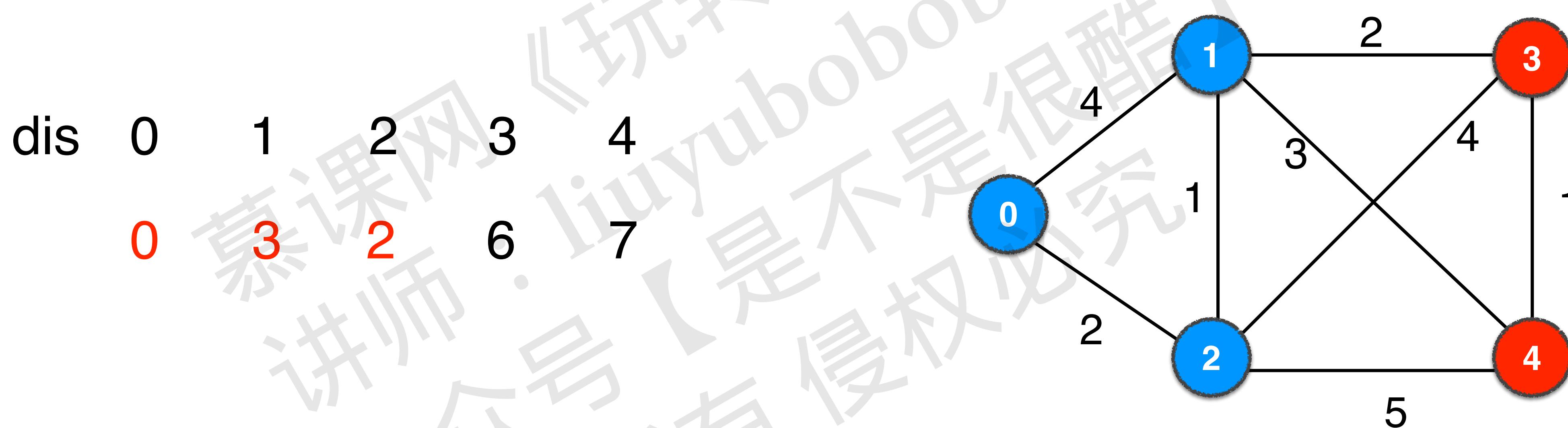
3

—

4

1

Dijkstra 算法



Dijkstra 算法

dis

2
2

3
5

17

0

卷之三

1

3

—

4

1

Dijkstra 算法

dis

2
2

3
5

6

0

卷之三

1

3

—

4

1

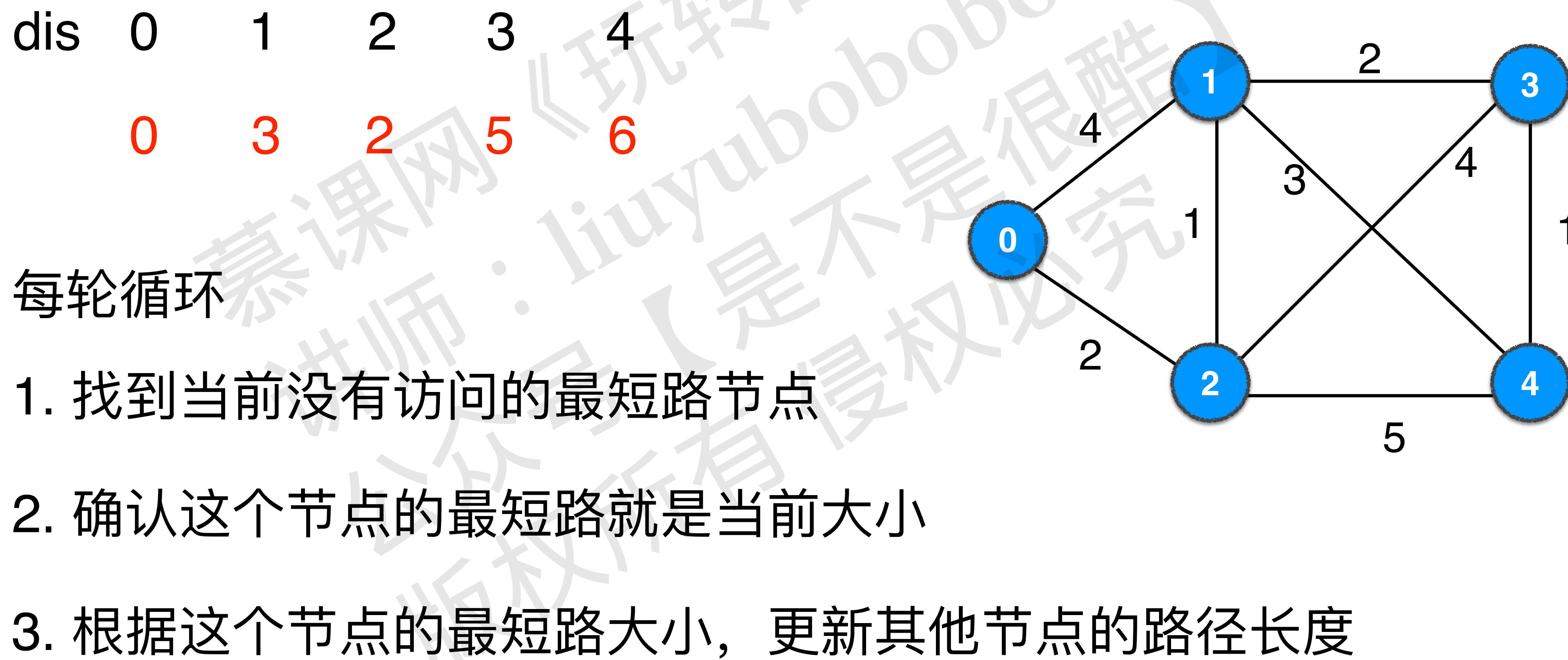
Dijkstra 算法



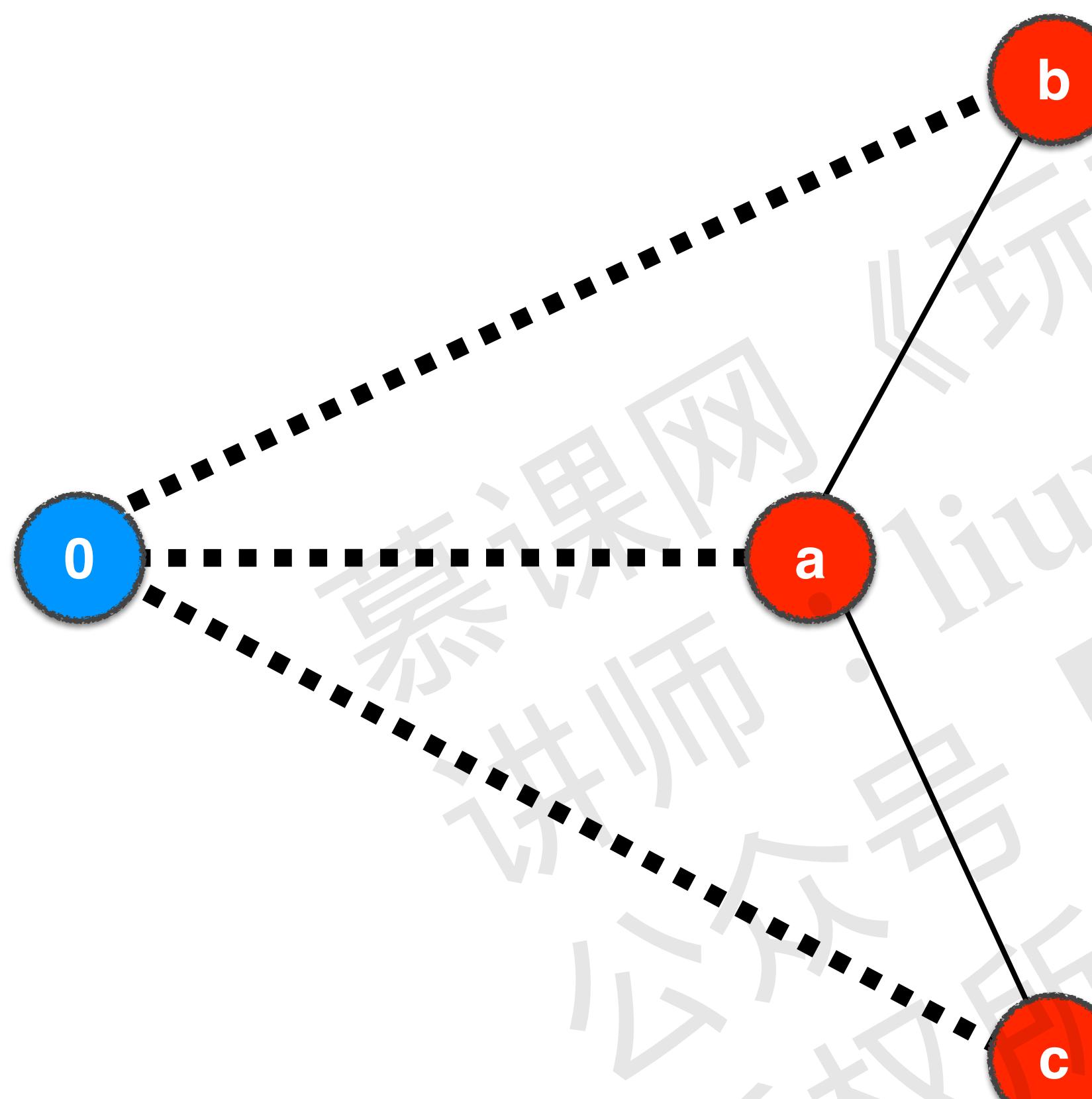
Dijkstra 算法



Dijkstra 算法

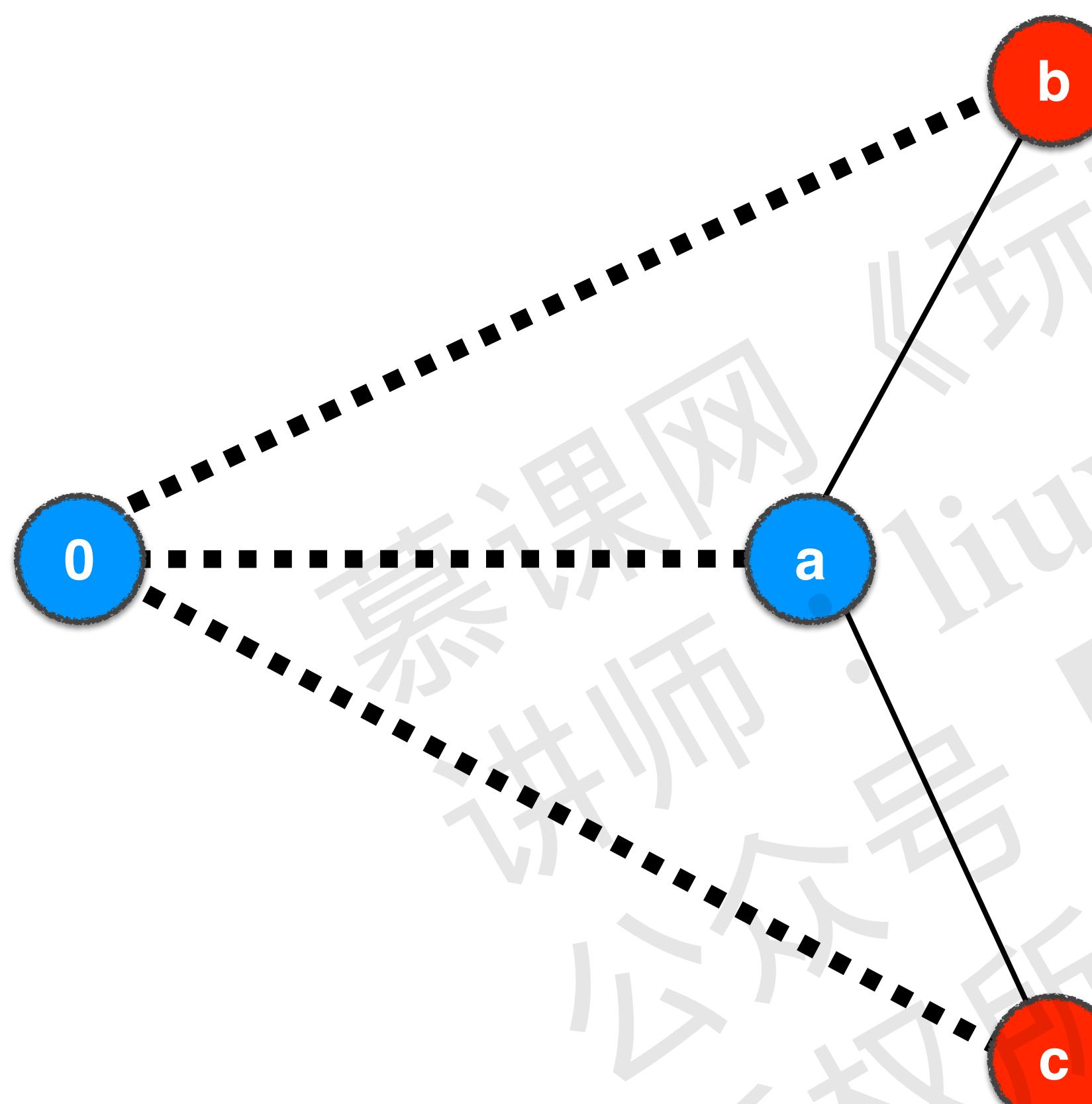


Dijkstra 算法



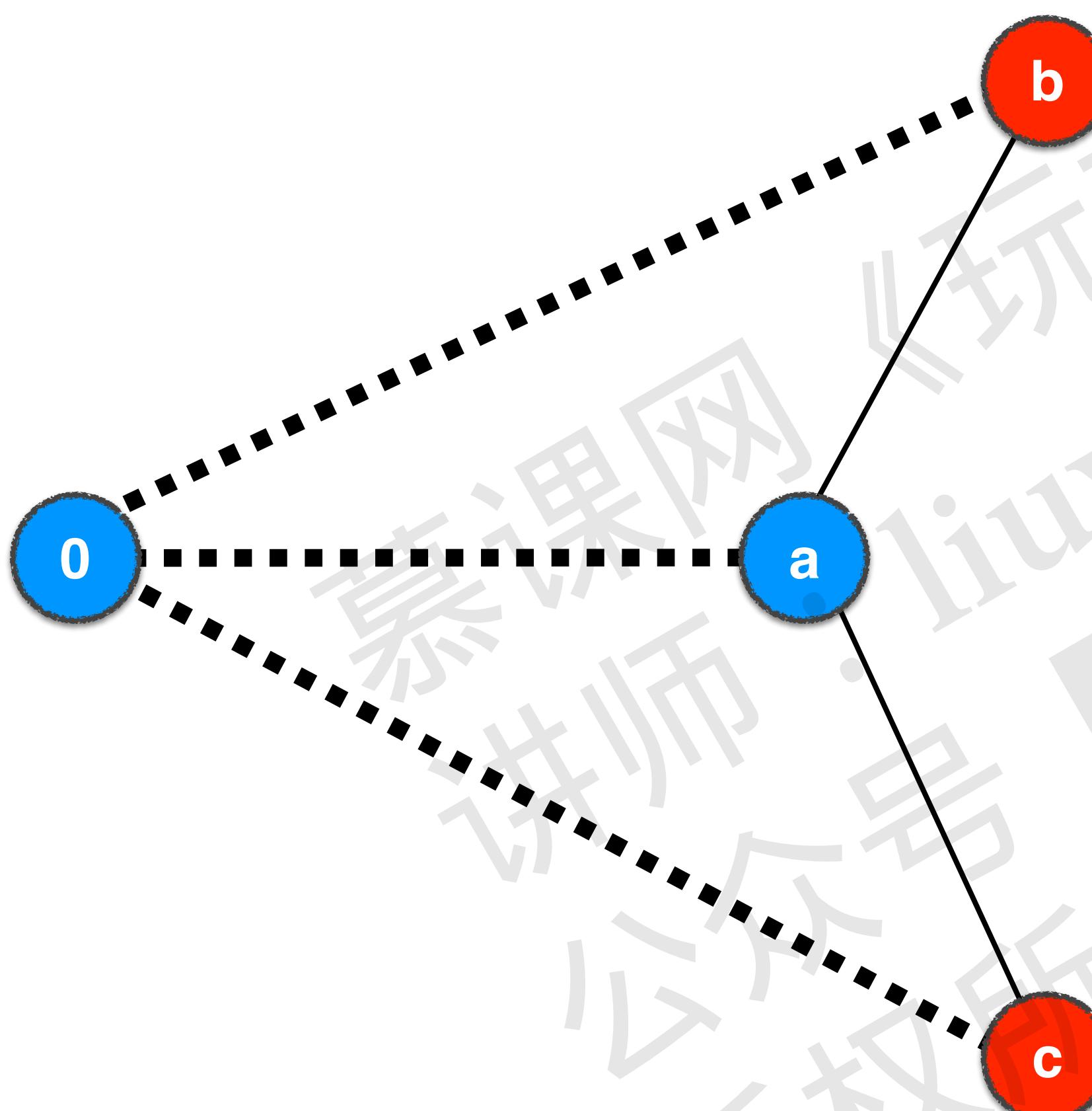
若 $\text{dis}[a]$ 最小,
则 $\text{dis}[a]$ 就是从源点到 a 的最短路径长度

Dijkstra 算法



若 $\text{dis}[a]$ 最小,
则 $\text{dis}[a]$ 就是从源点到 a 的最短路径长度

Dijkstra 算法

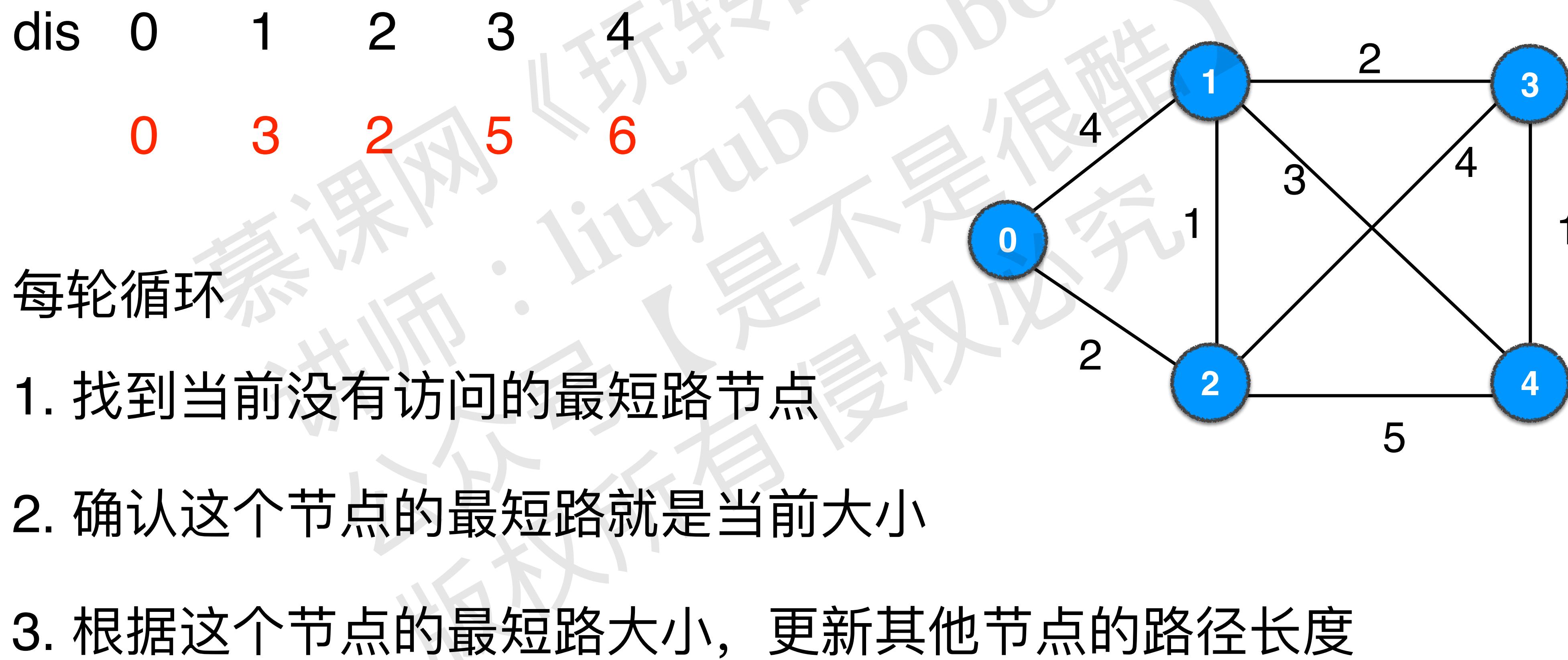


若 $\text{dis}[a]$ 最小,
则 $\text{dis}[a]$ 就是从源点到 a 的最短路径长度

$$\text{dis}[b] = \min(\text{dis}[b], \text{dis}[a] + ab)$$

$$\text{dis}[c] = \min(\text{dis}[c], \text{dis}[a] + ac)$$

Dijkstra 算法



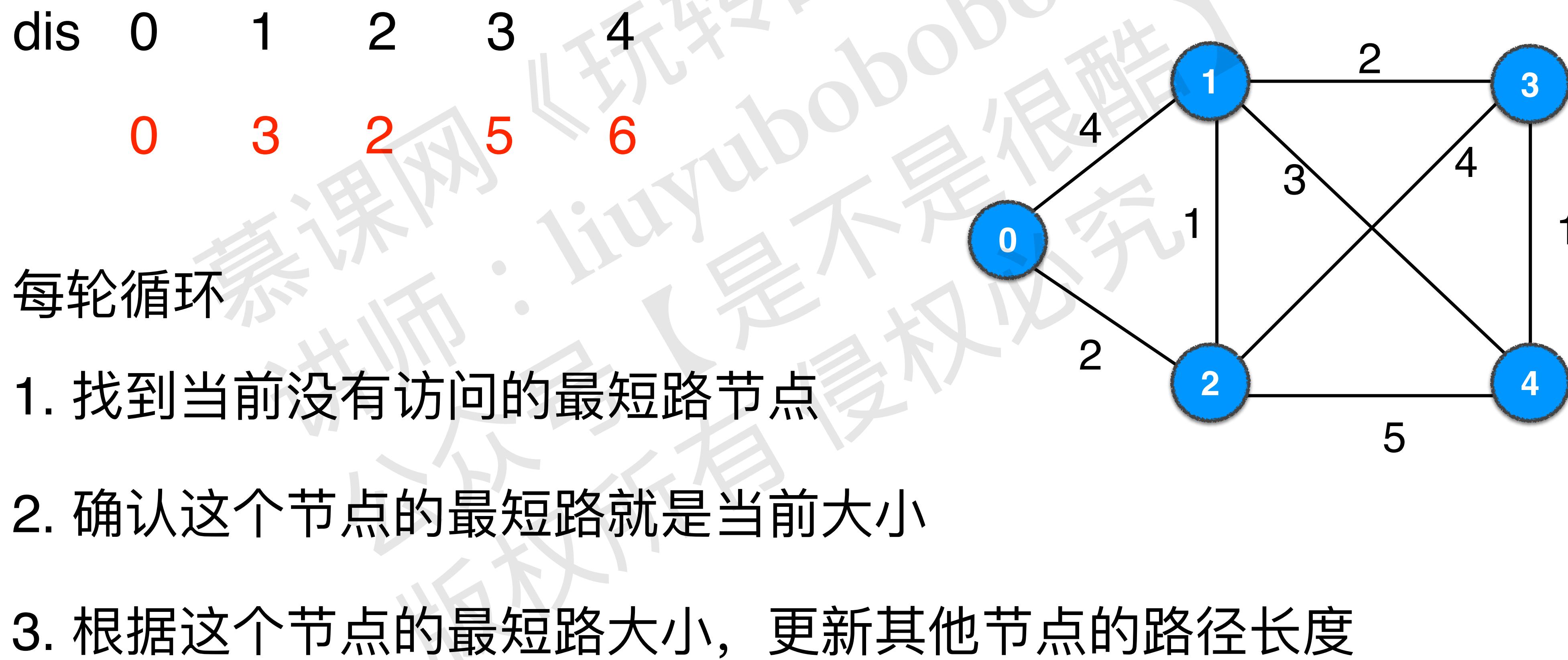
实现 Dijkstra 算法

liuyubobobo

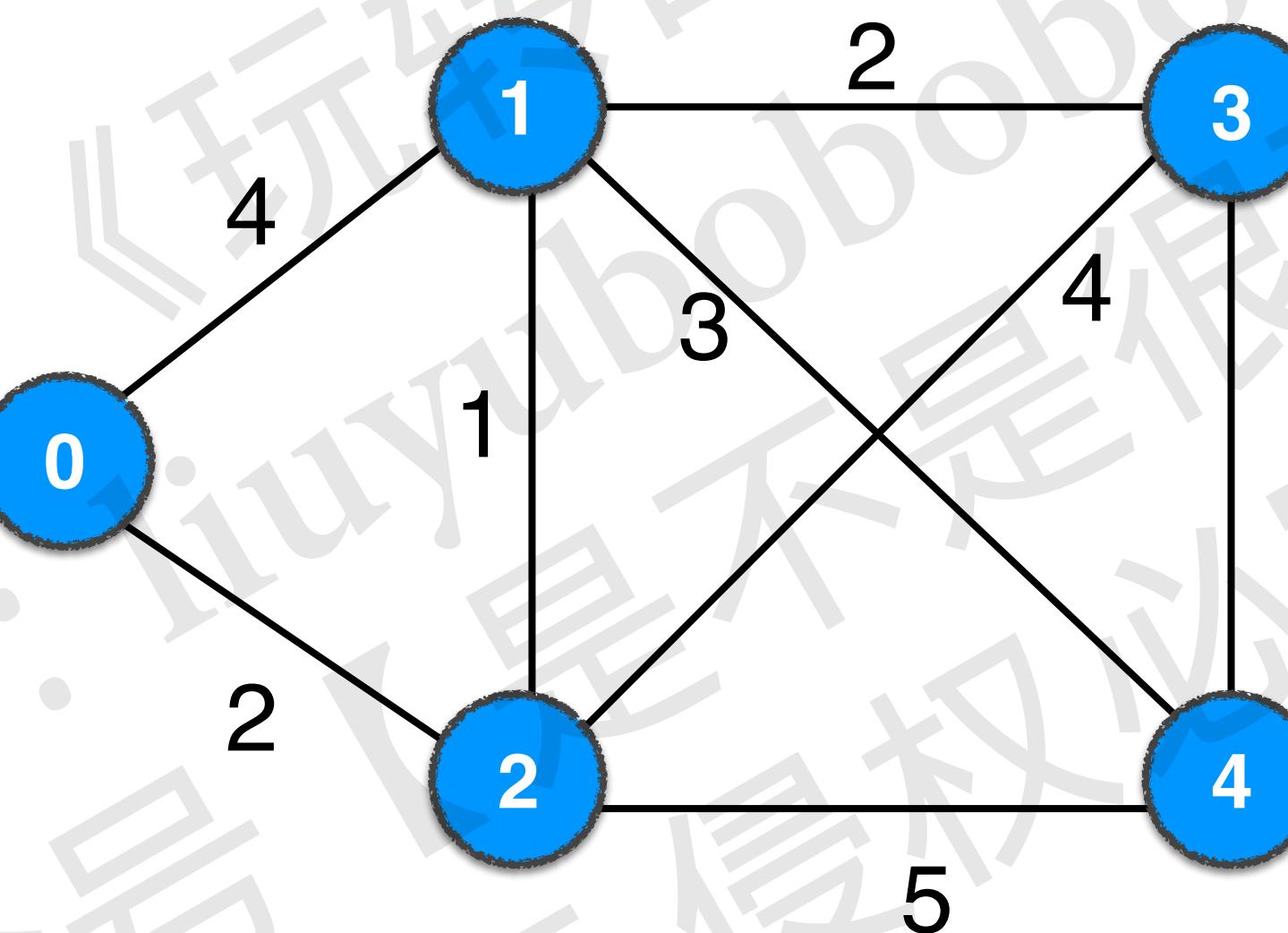
编程实践：实现 Dijkstra 算法

慕课网 · liuyibobobo · 编程实践：实现 Dijkstra 算法

Dijkstra 算法



Dijkstra 算法



$O(V^2)$

Dijkstra 算法的优化

liuyubobobo

Dijkstra 算法的优化

```
while(true){  
  
    int cur = -1, curdis = Integer.MAX_VALUE;  
    for(int v = 0; v < G.V(); v++)  
        if(!visited[v] && dis[v] < curdis){  
            curdis = dis[v];  
            cur = v;  
        }  
  
    if(cur == -1) break;  
  
    visited[cur] = true;  
    for(int w: G.adj(cur))  
        if(!visited[w] && dis[cur] + G.getWeight(cur, w) < dis[w])  
            dis[w] = dis[cur] + G.getWeight(cur, w);  
}
```

瓶颈

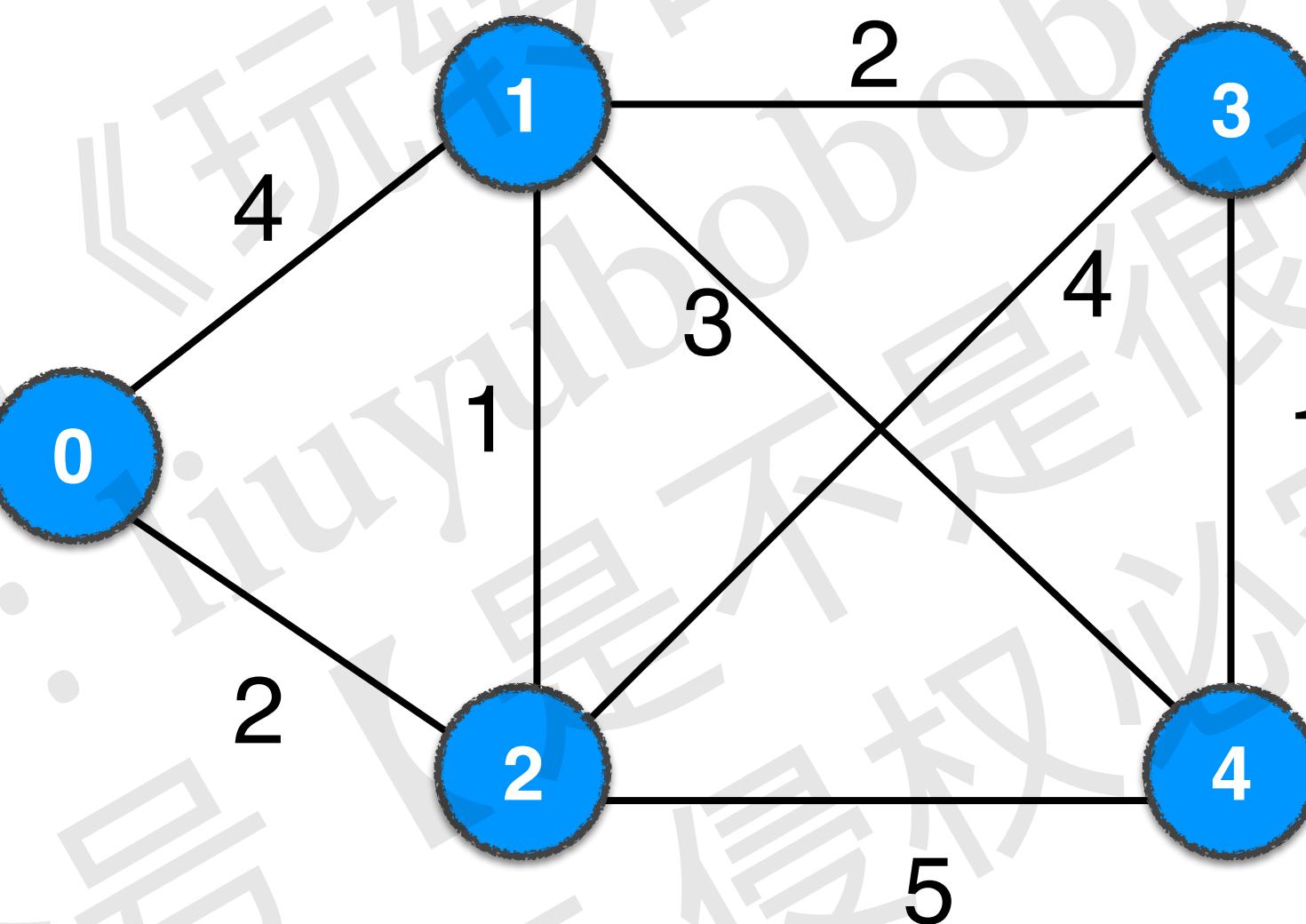
找到当前未访问的
dis 值最小的节点

优先队列

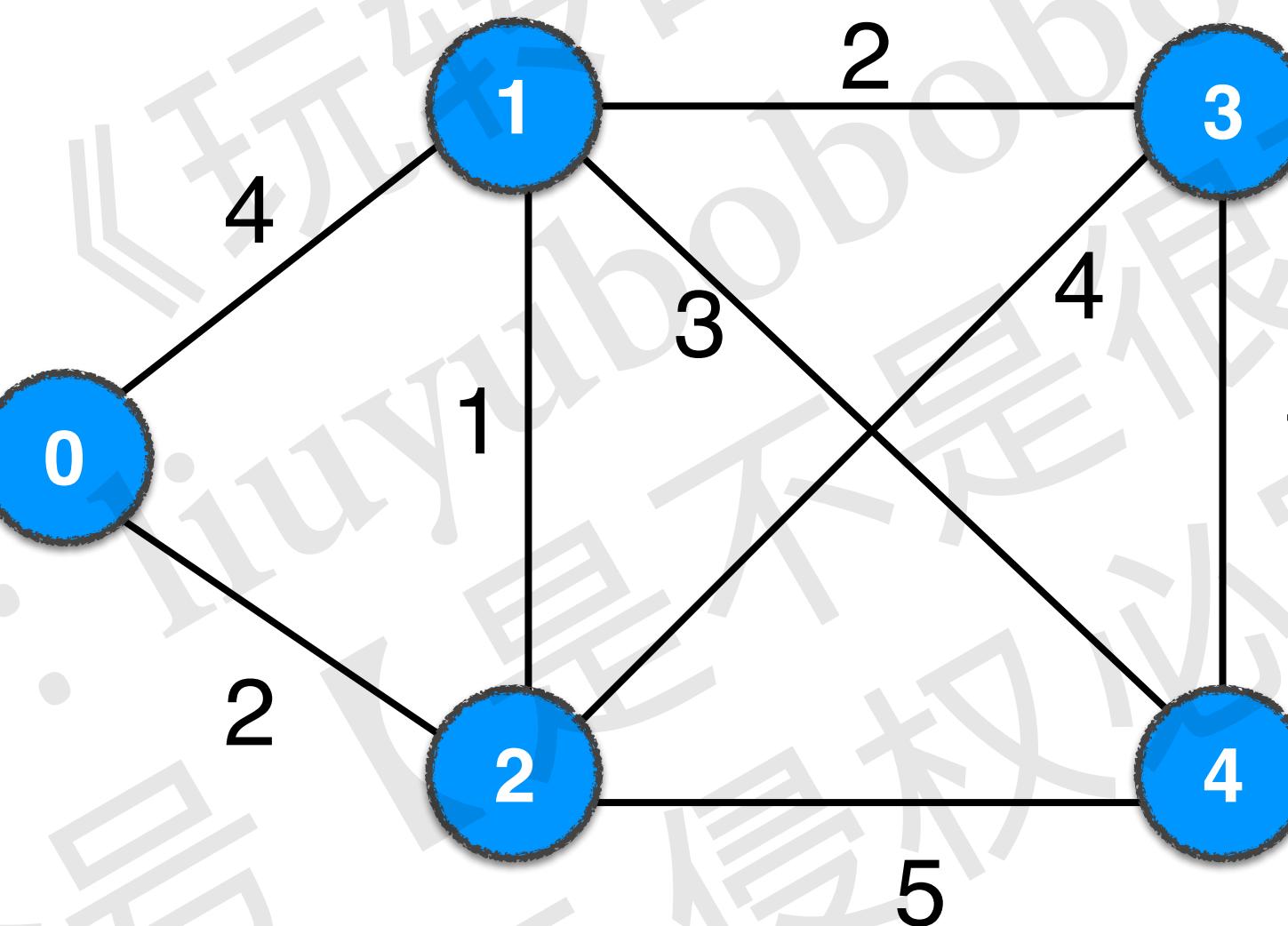
编程实践：优化 Dijkstra 算法

慕课网 · liuyibobobo · 《玩转图论算法》

Dijkstra 算法



Dijkstra 算法



$O(E \log E)$

更多 Dijkstra 算法相关的讨论

liuyubobobo

更多 Dijkstra 算法相关的讨论

求解最短路径

pre 数组

dis

0 3 2

1 6

2 7

3

4

5

6

7



更多 Dijkstra 算法相关的讨论

求解最短路径

pre 数组

dis

0 3 2

1 6

2 7

3

4

5

6

7



更多 Dijkstra 算法相关的讨论

求解最短路径

pre 数组

dis

0

1

2

3

4

5

7

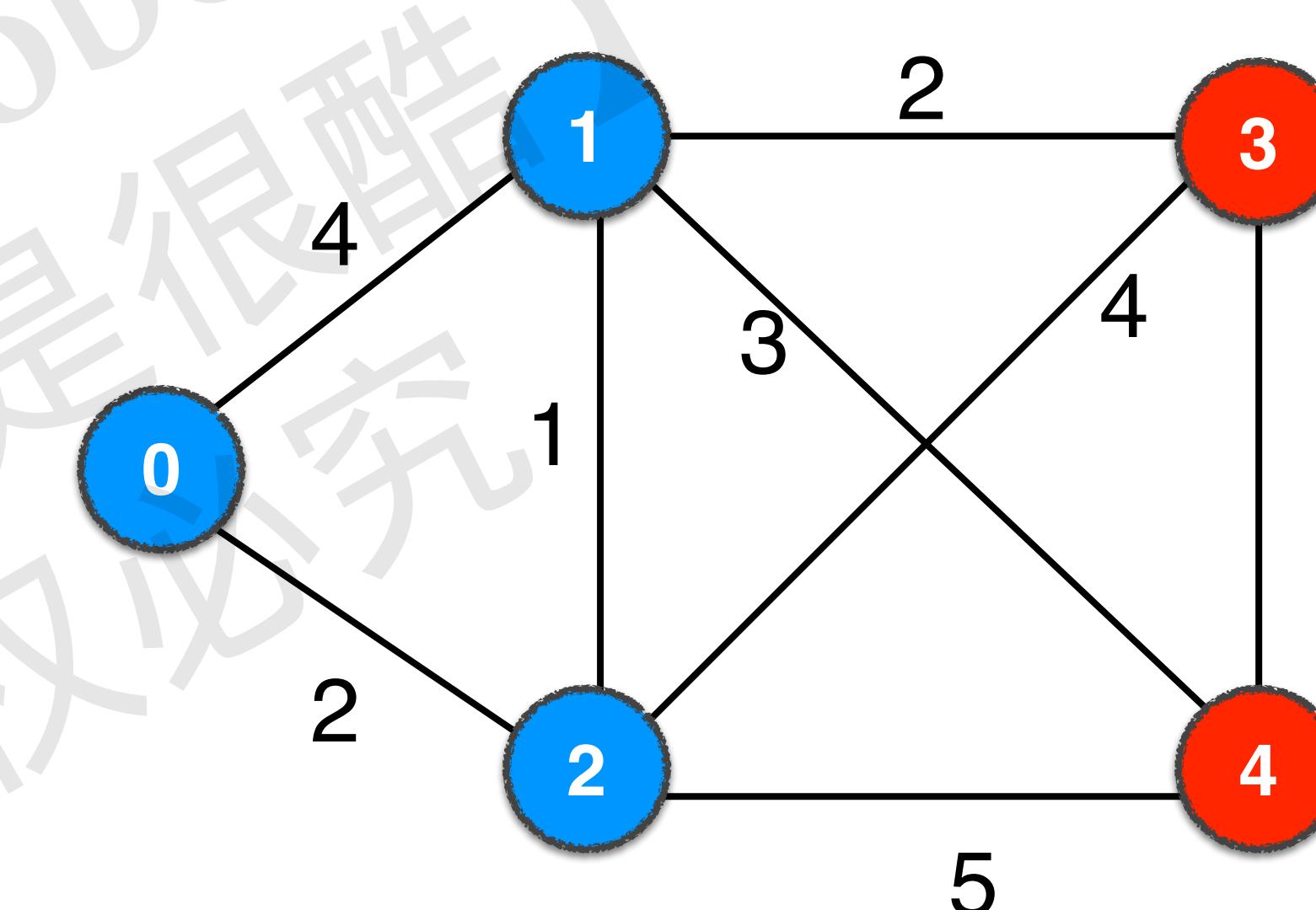
0

3

2

5

7



更多 Dijkstra 算法相关的讨论

求解最短路径

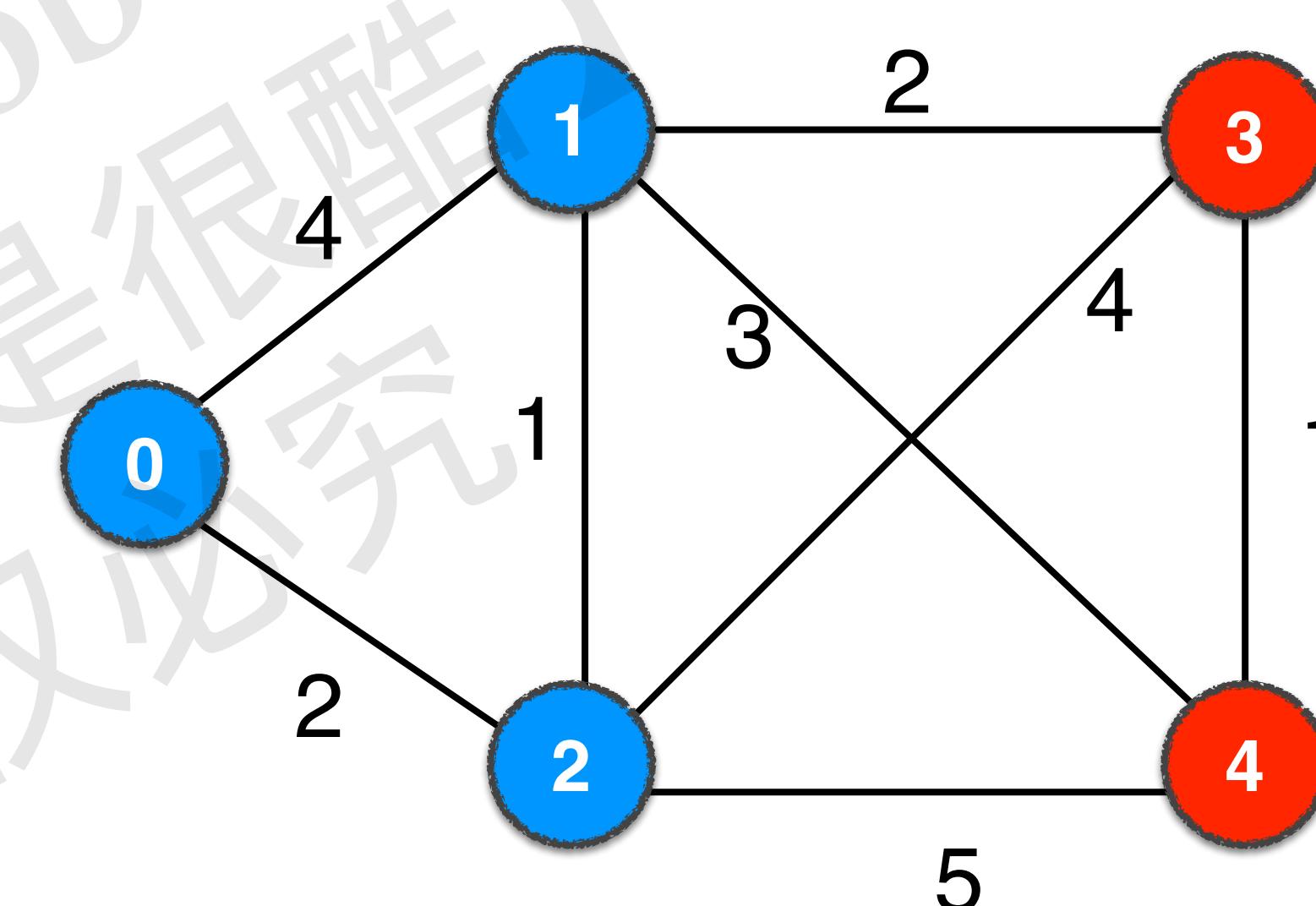
pre 数组

dis

0 1 2 3 4 5 7

0 3 2 5

pre [3] = 1



更多 Dijkstra 算法相关的讨论

求解最短路径

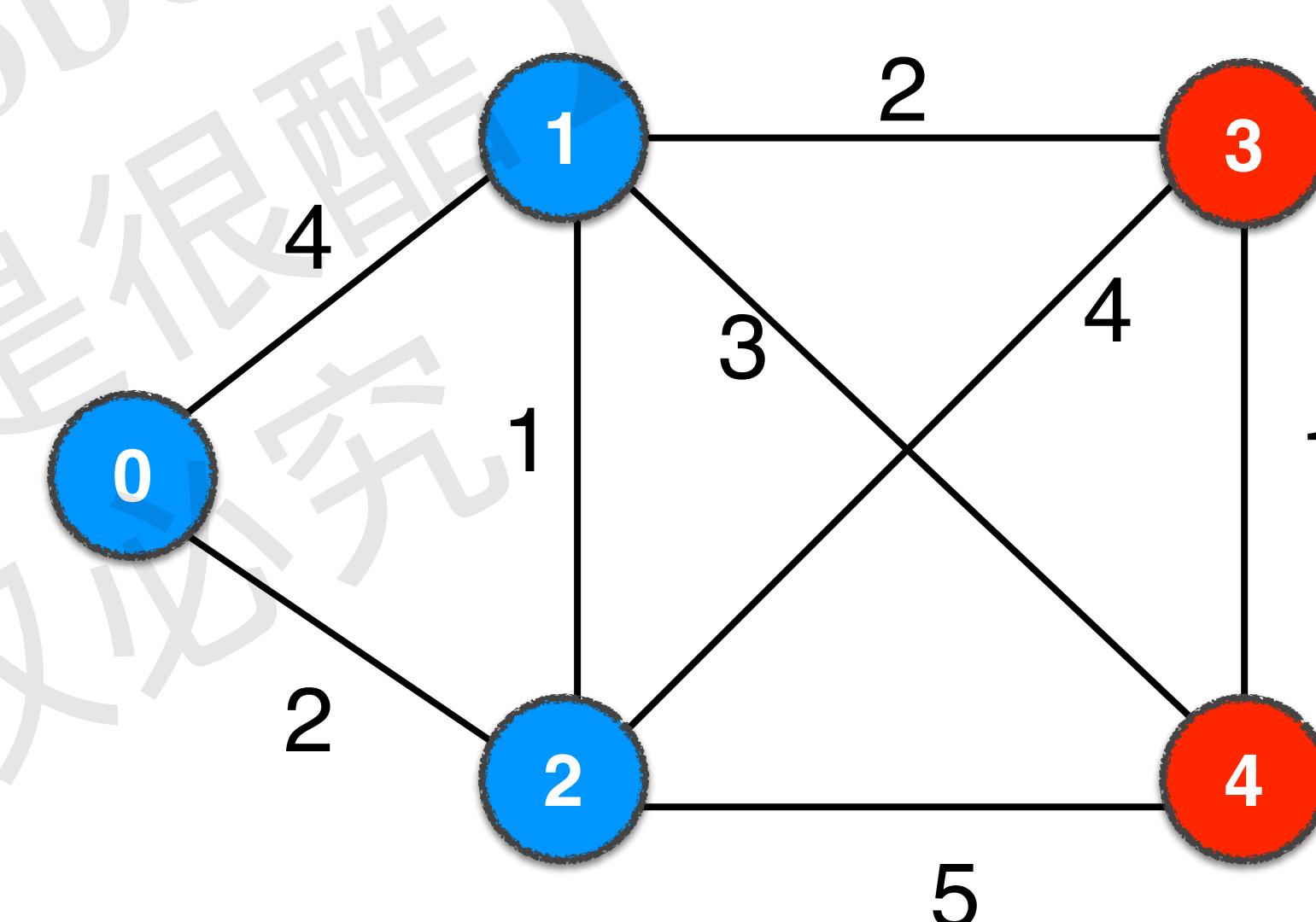
pre 数组

dis

0 1 2 3 4
0 3 2 5 6

pre [3] = 1

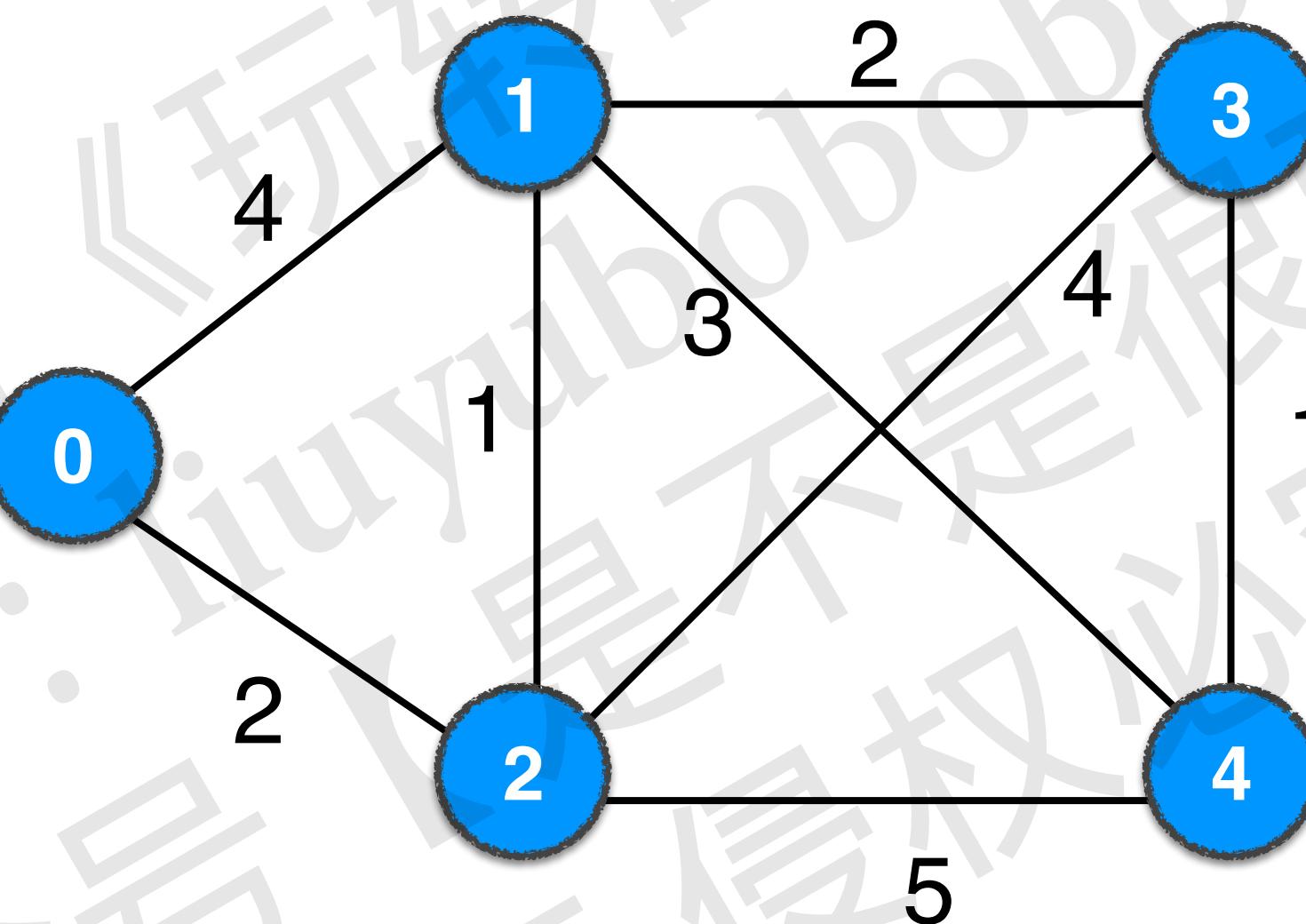
pre [4] = 1



编程实践：求解最短路径

编程实践：求解最短路径

Dijkstra 算法



更多 Dijkstra 算法相关的讨论

时间复杂度： $E \log E$

可以优化到： $E \log V$ 借助索引堆

The screenshot shows a course page with the following details:

- Category: / 实战 / 算法与数据结构--综合提升篇 (c++版)
- 收藏 (Collection): Star icon
- 分享 (Share): Icons for WeChat, Weibo, and QQ
- Title: 算法与数据结构--综合提升篇 (c++版)
- Description: 口碑人气双高的bobo老师算法课，让算法变得简单好学，真实好评看的到
- Price: ¥ 166.00
- Payment Options: 花呗付款 (Huabei Payment), 京东白条 (JD Baijie), 可用优惠券 (Available Coupon)
- Difficulty: 中级 (Intermediate)
- Duration: 时长 13小时35分钟 (Duration 13 hours 35 minutes)
- Student Count: 学习人数 8701 (Number of Students 8701)
- Review Rate: 好评度 99.9% (Good Review Rate 99.9%)
- Enter Course (进入课程) button

Dijkstra 算法

只关注从 s 到 t 之间的最短路径?

所有点对最短路径?

运行 V 次 Dijkstra 算法。 $V^* E \log E$

溢出问题

更多 Dijkstra 算法相关的讨论

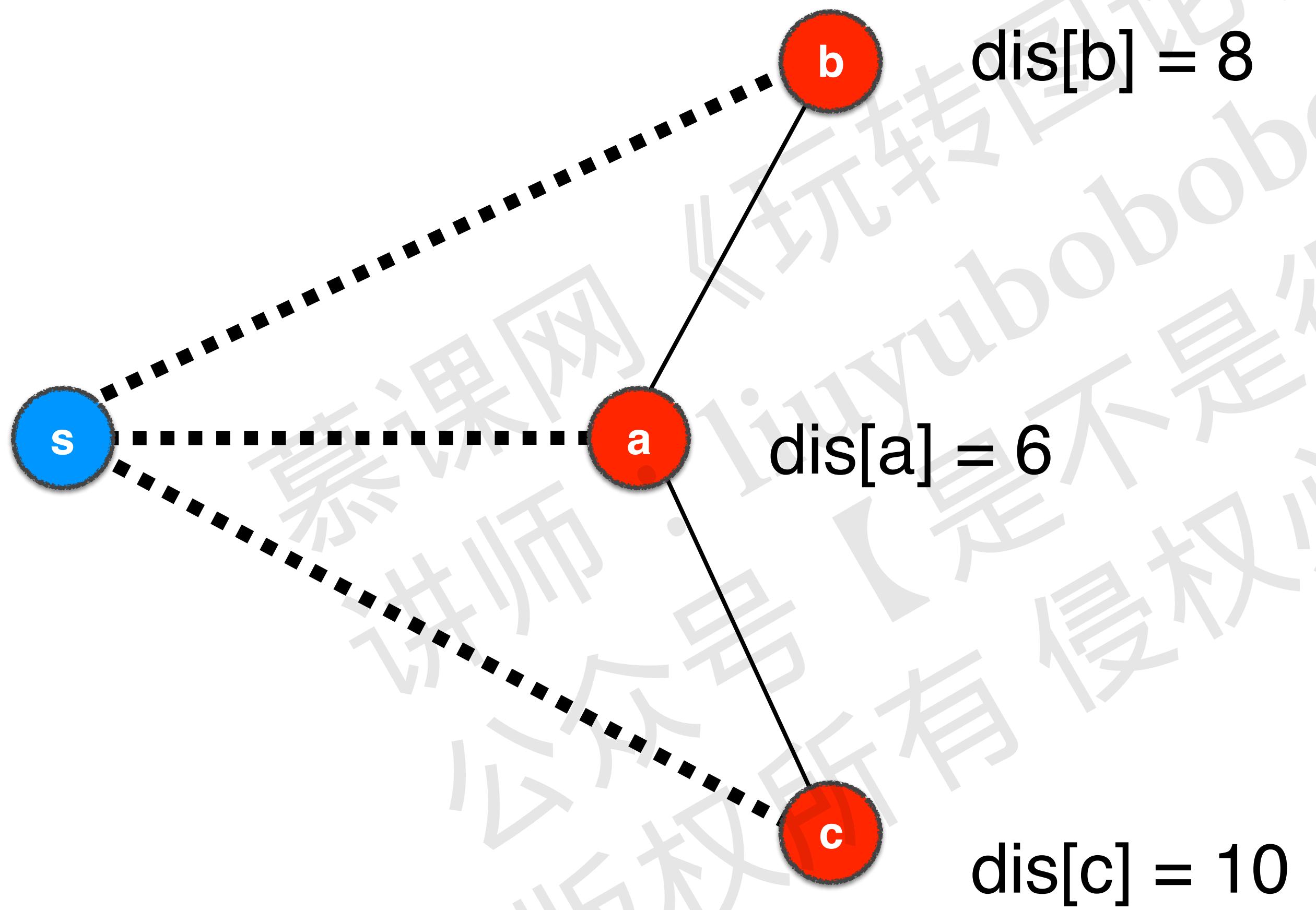
Dijkstra: 不能处理负权边

处理负权边: Bellman-Ford

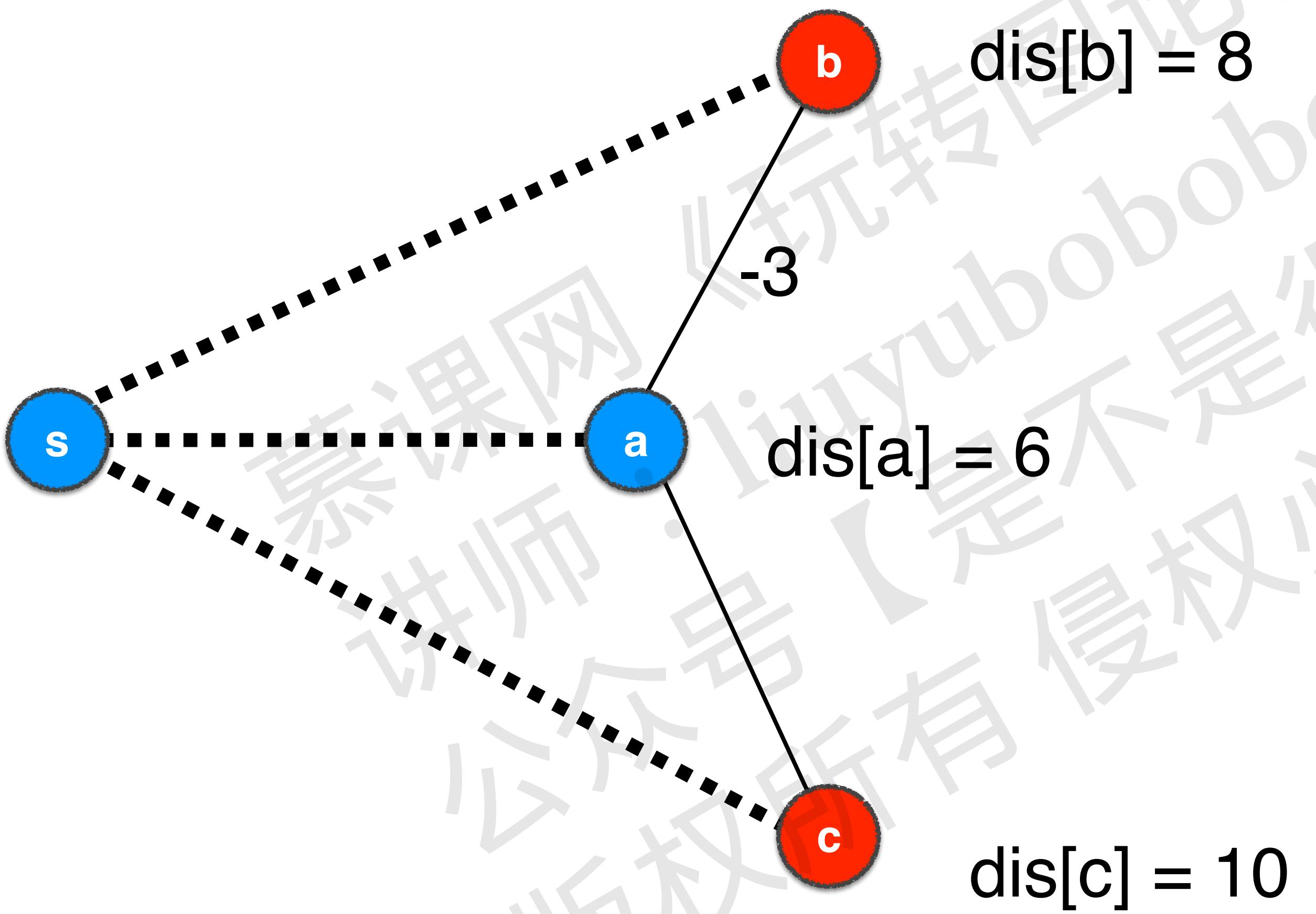
Bellman-Ford 算法

liuyubobobo

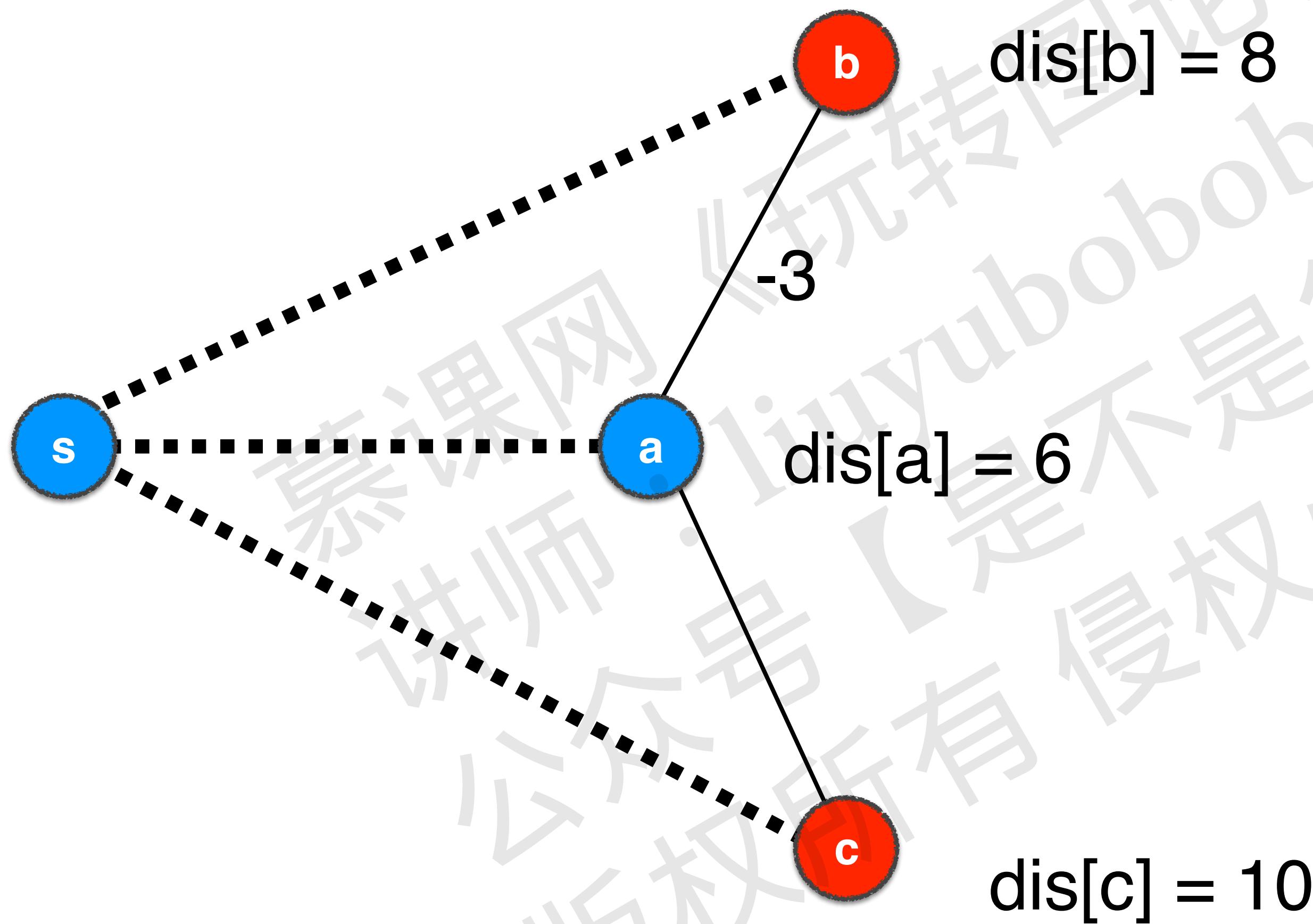
Bellman-Ford 算法



Bellman-Ford 算法



Bellman-Ford 算法



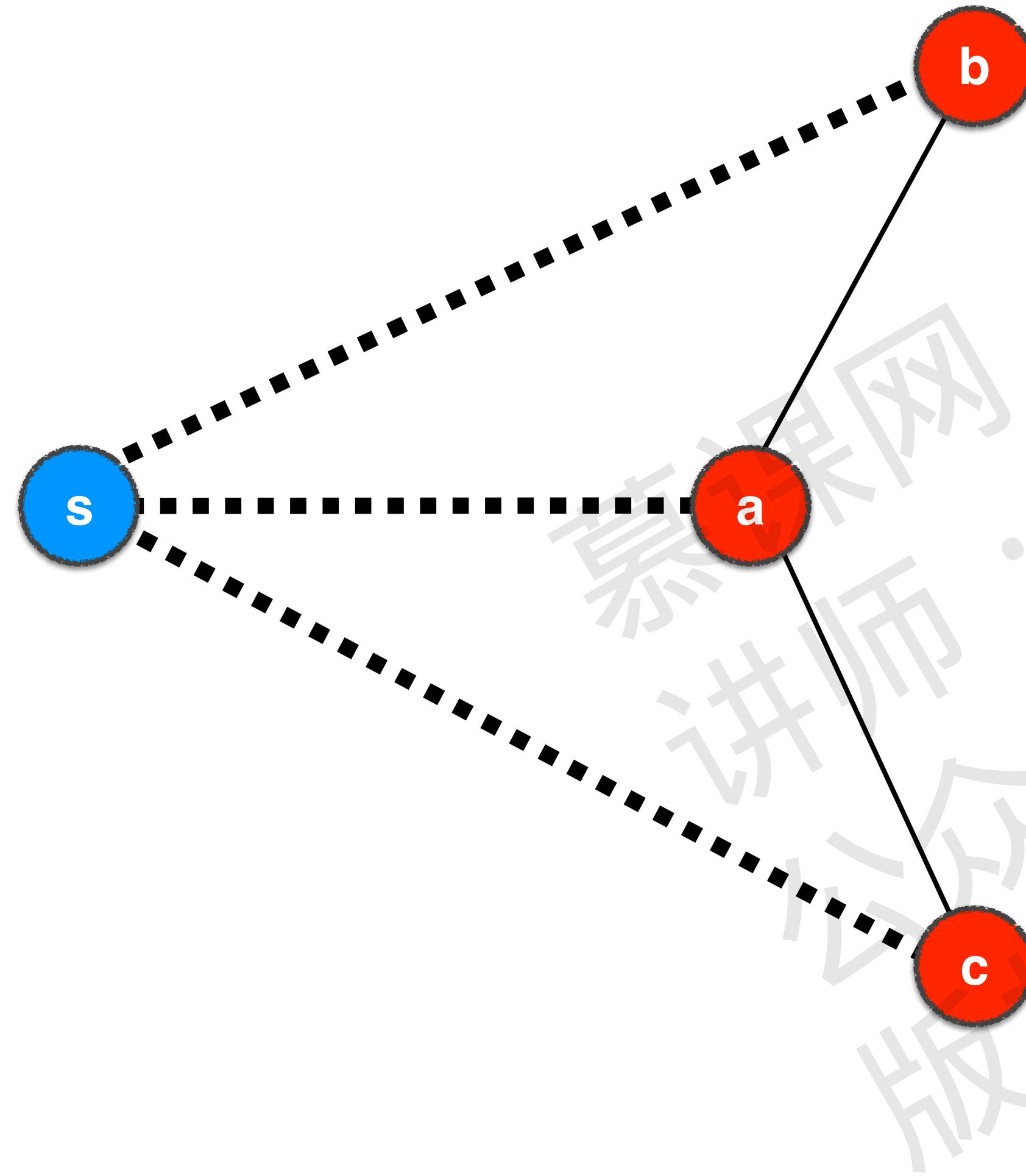
if ($dis[b] + ba < dis[a]$)
 $dis[a] = dis[b] + ba$

if ($dis[a] + ab < dis[b]$)
 $dis[b] = dis[a] + ab$

松弛操作 Relaxation

这个过程和边的权值的正负无关

松弛操作



假设 $\text{dis}[v]$ 是从 s 到 v 的经过边数不超过 k 的最短距离

```
if ( $\text{dis}[a] + ab < \text{dis}[b]$ )  
     $\text{dis}[b] = \text{dis}[a] + ab$ 
```

找到从 s 到 b 的经过边数不超过 $k + 1$ 的最短距离

```
if ( $\text{dis}[b] + ba < \text{dis}[a]$ )  
     $\text{dis}[a] = \text{dis}[b] + ba$ 
```

找到从 s 到 a 的经过边数不超过 $k + 1$ 的最短距离

Bellman-Ford 算法

Bellman-Ford 算法：

初始 $dis[s] = 0$, 其余 dis 值为 ∞

对所有边进行一次松弛操作，则求出了到所有点，经过的边数最多为1的最短路

对所有边再进行一次松弛操作，则求出了到所有点，经过的边数最多为2的最短路

...

对所有边进行 $V - 1$ 次松弛操作，则求出了到所有点，经过的边数最多为 $V - 1$ 的最短路

负权环

liuyubobo

慕课网《玩转图论算法》
讲师：liuyubobo
公众账号：负权环
版权所有 © liuyubobo

Bellman-Ford 算法

Bellman-Ford 算法：

初始 $dis[s] = 0$, 其余 dis 值为 ∞

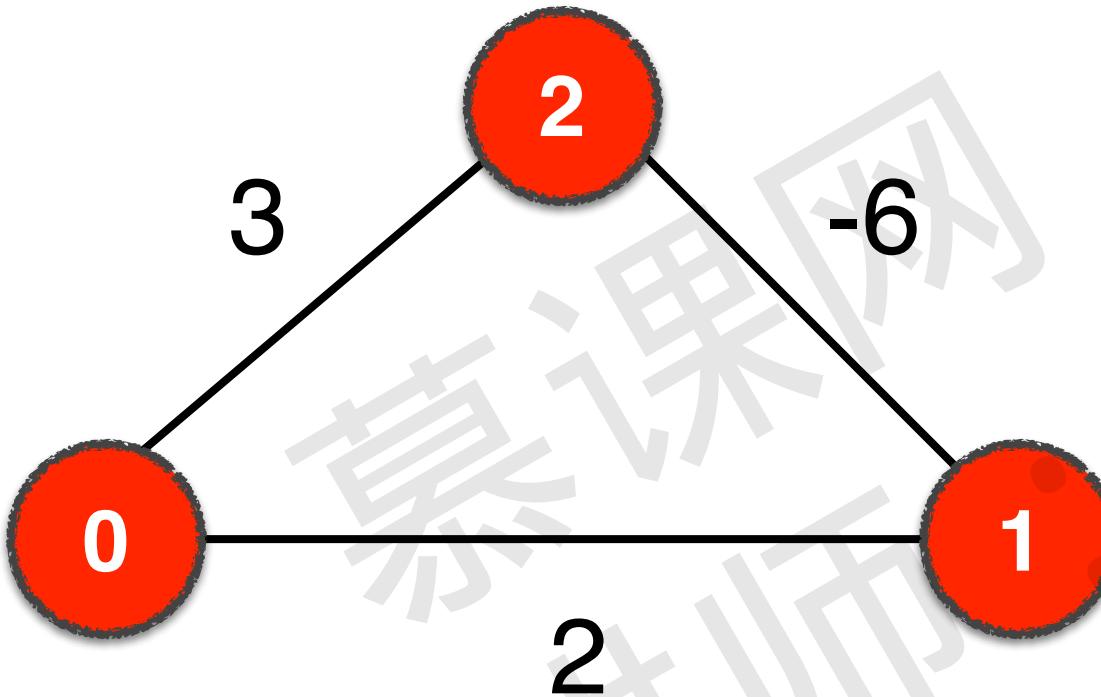
对所有边进行一次松弛操作，则求出了到所有点，经过的边数最多为1的最短路

对所有边再进行一次松弛操作，则求出了到所有点，经过的边数最多为2的最短路

...

对所有边进行 $V - 1$ 次松弛操作，则求出了到所有点，经过的边数最多为 $V - 1$ 的最短路

Bellman-Ford 算法

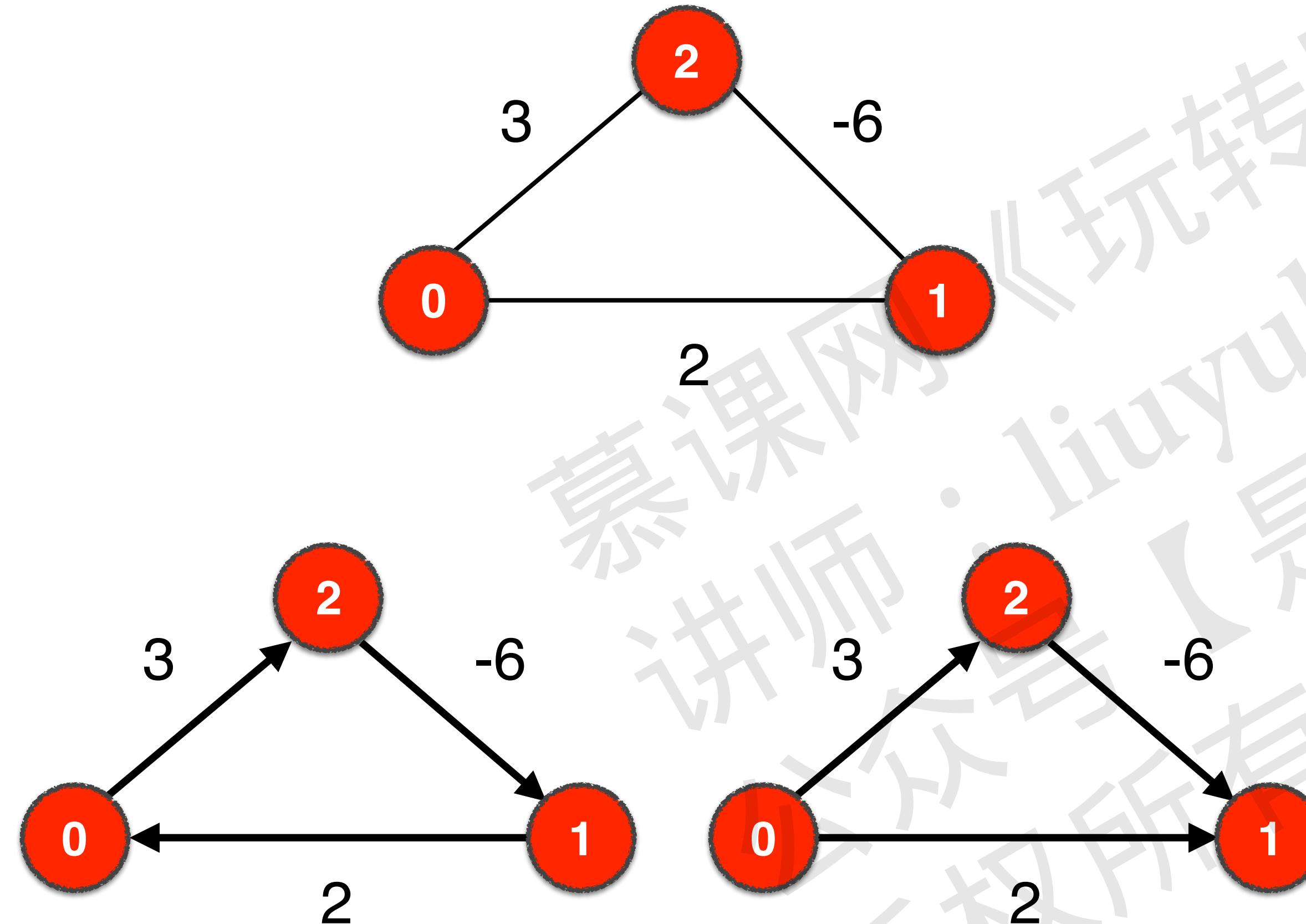


负权环

当存在负权环的时候，没有最短路径

对于无向图，有负权边等同于有负权环

Bellman-Ford 算法



负权环

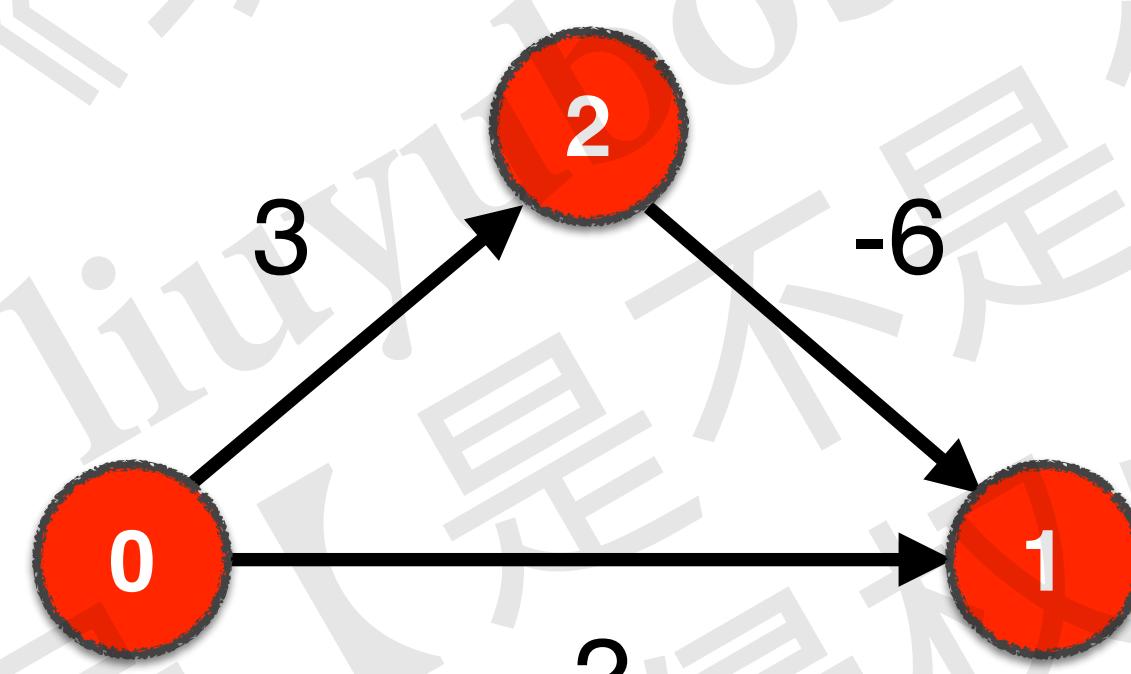
当存在负权环的时候，没有最短路径

对于无向图，有负权边等同于有负权环

模拟1

慕课网《玩转图论算法》
讲师：liuyubobobo
公众版所有侵权必究
版权所有人是很酷

Bellman-Ford 算法



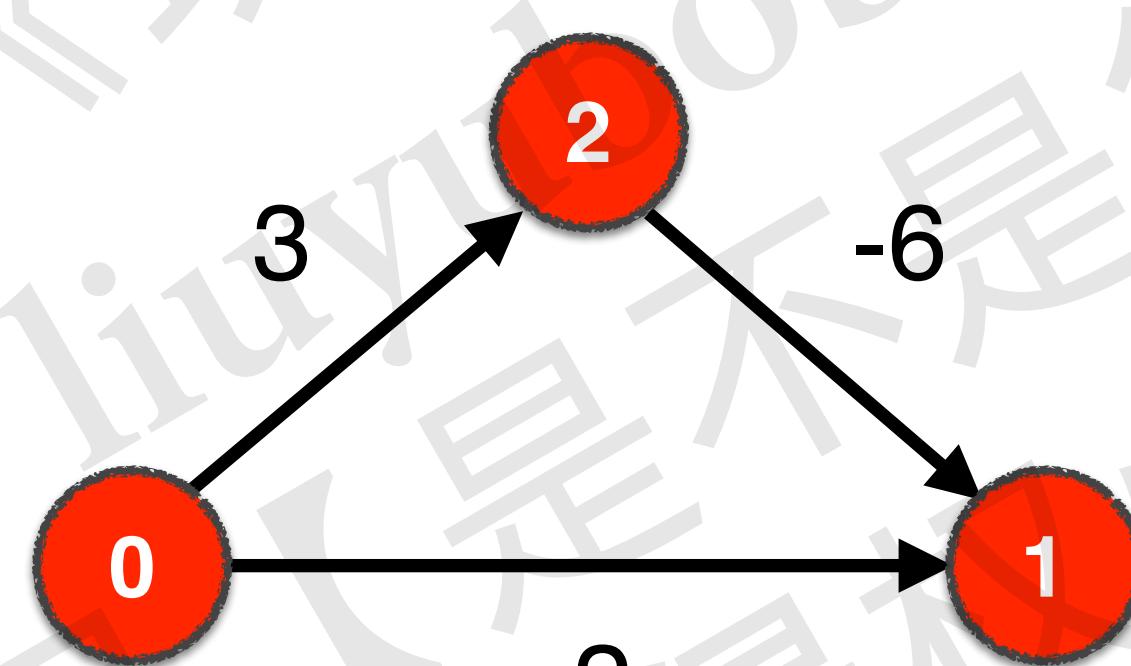
$dis[2] = \infty$

$relax(0, 1)$

$dis[0] = 0$

$dis[1] = \infty$

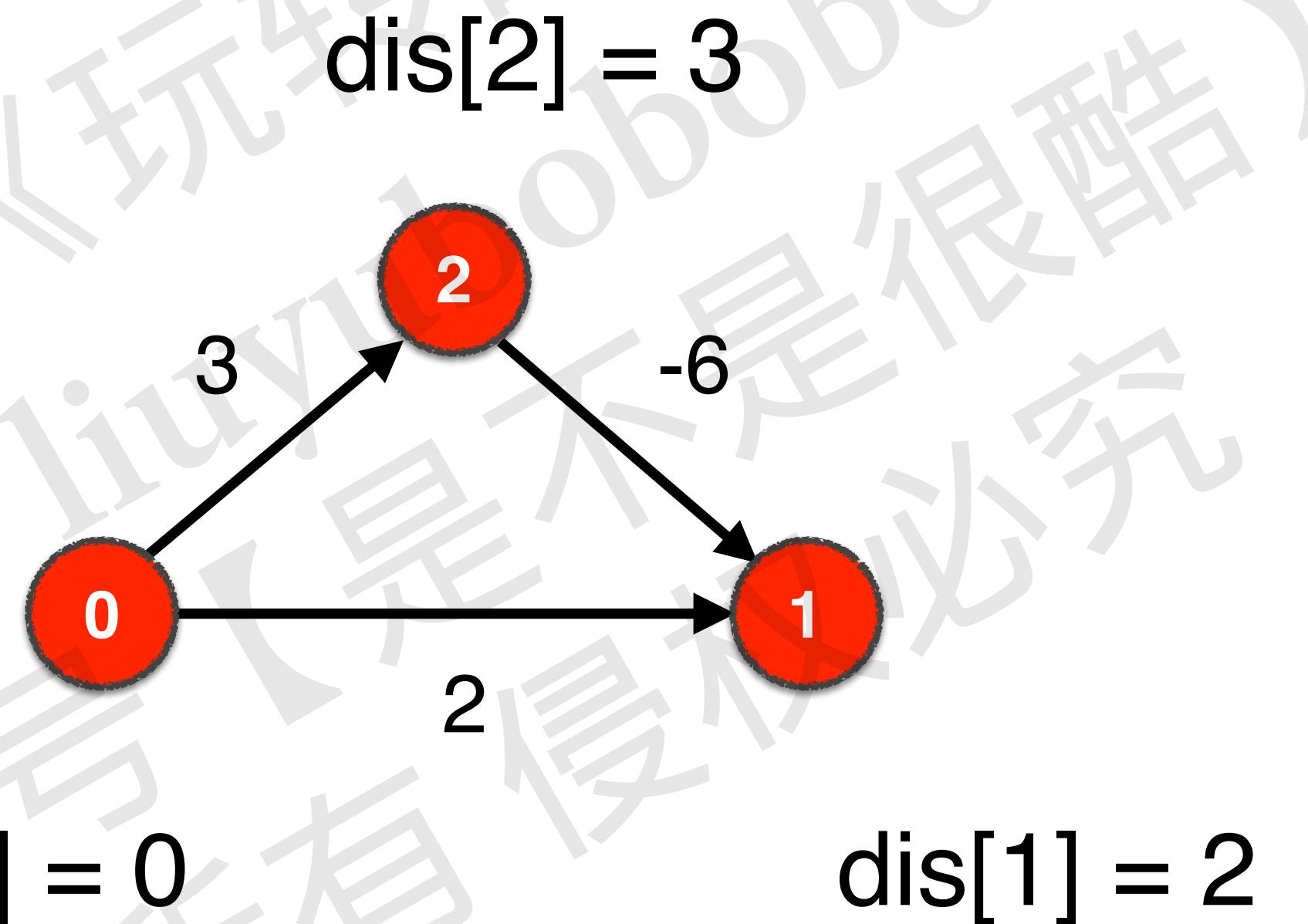
Bellman-Ford 算法



relax(0, 1)

relax(0, 2)

Bellman-Ford 算法



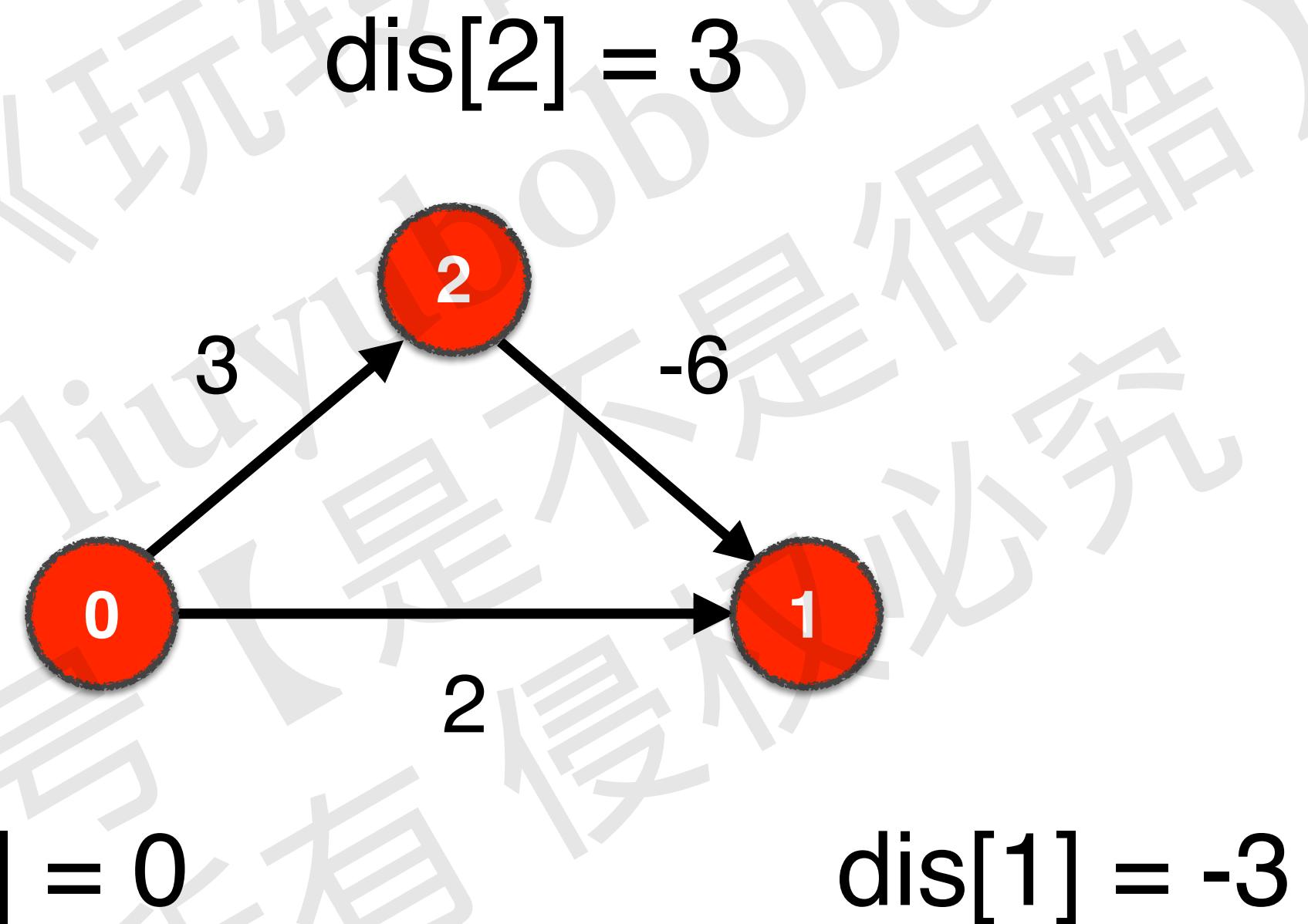
$\text{relax}(0, 1)$

$\text{relax}(0, 2)$

$\text{relax}(2, 1)$

$\text{dis}[1] = 2$

Bellman-Ford 算法

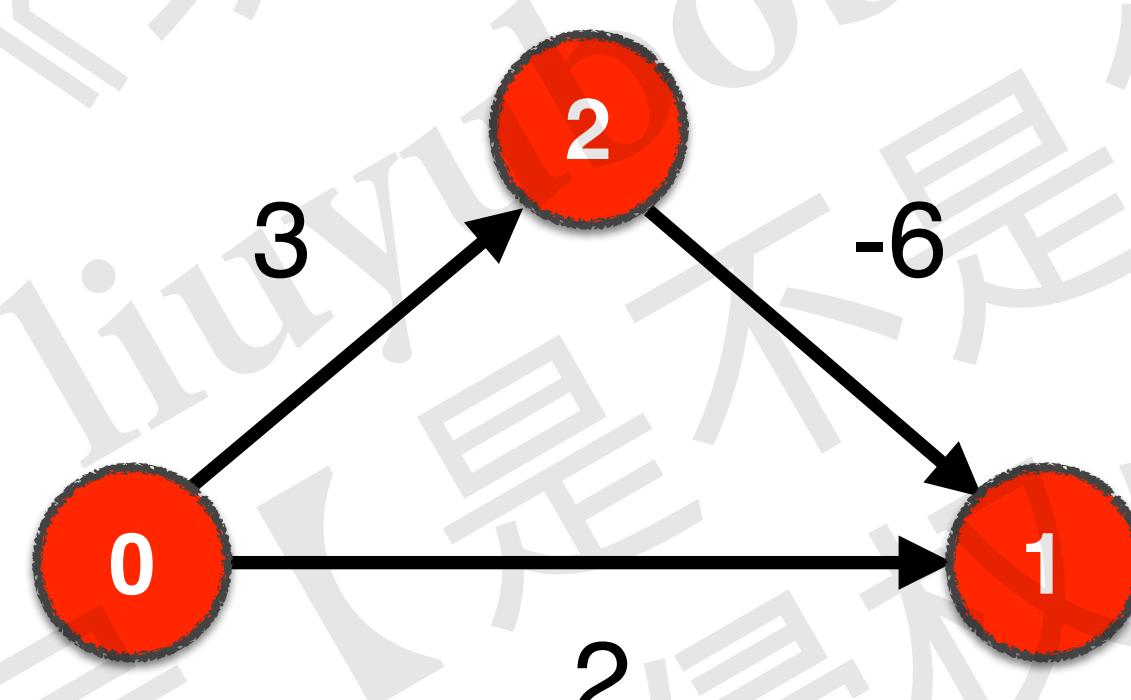


relax(0, 1)
relax(0, 2)
relax(2, 1)

模拟2

慕课网《玩转图论算法》
讲师：liuyubobobo
公众账号：模拟人是很酷
版权所有 © 侵权必究

Bellman-Ford 算法



$dis[2] = \infty$

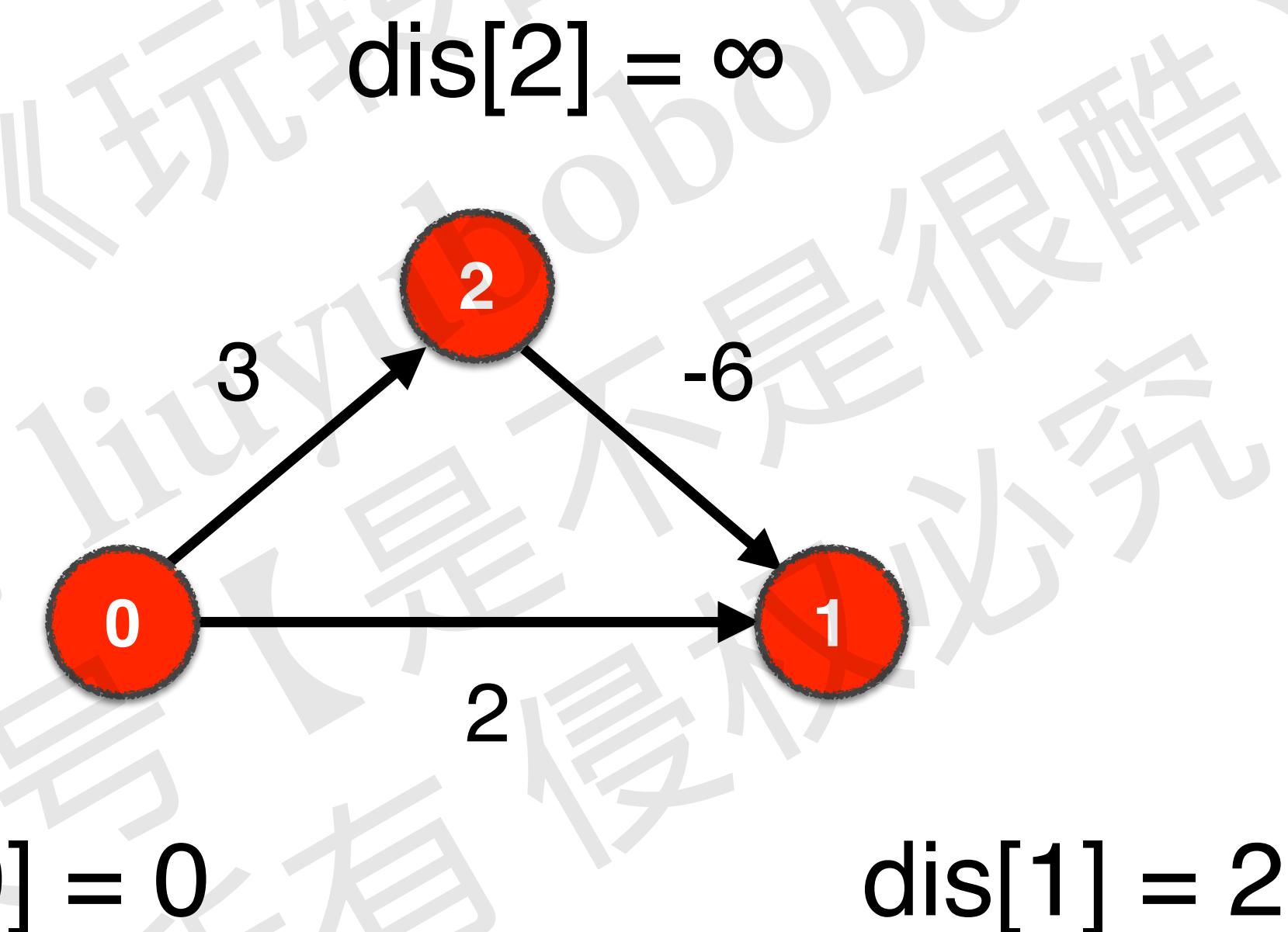
$dis[0] = 0$

$dis[1] = \infty$

$relax(2, 1)$

$relax(0, 1)$

Bellman-Ford 算法

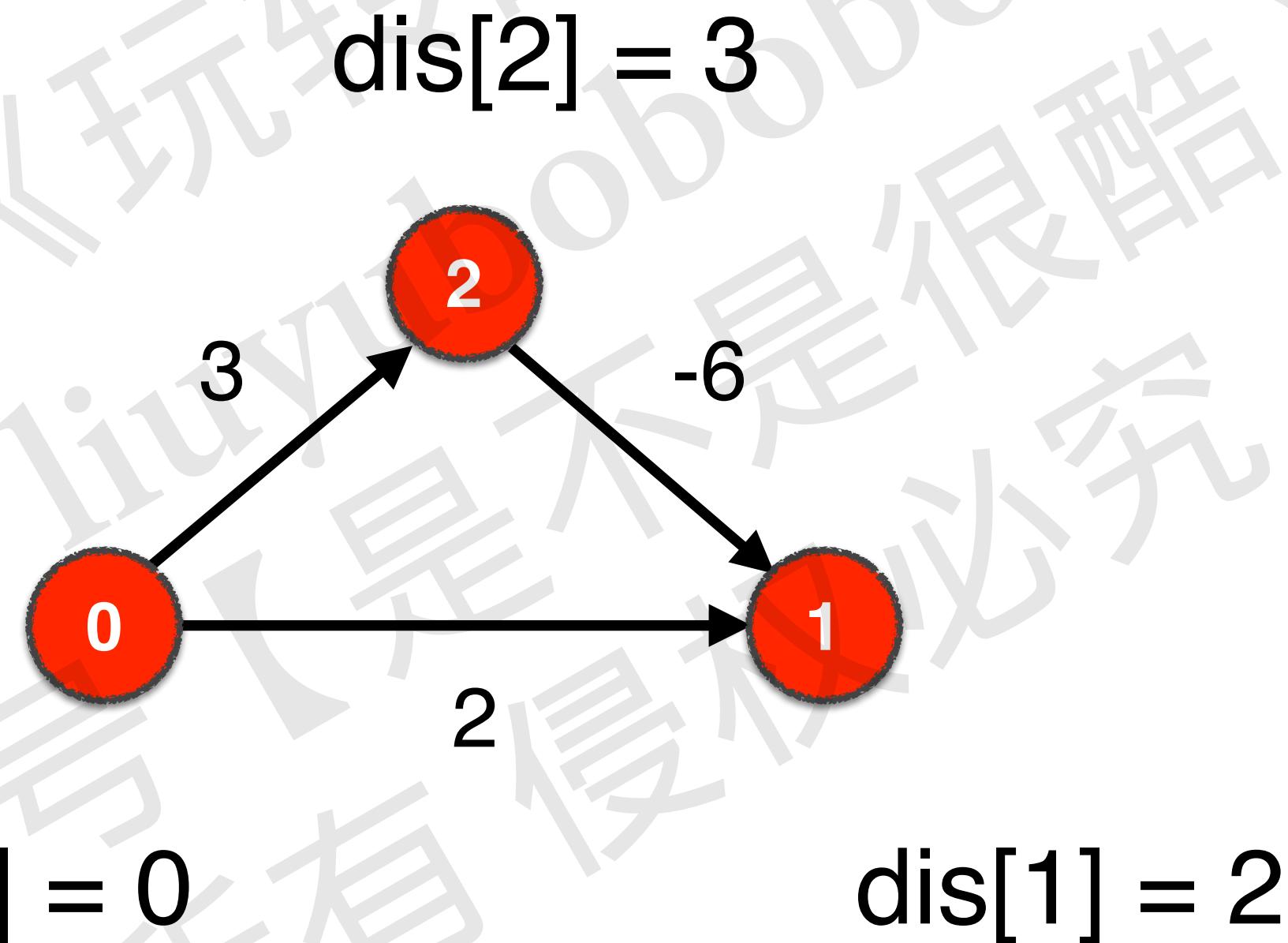


relax(2, 1)

relax(0, 1)

relax(0, 2)

Bellman-Ford 算法



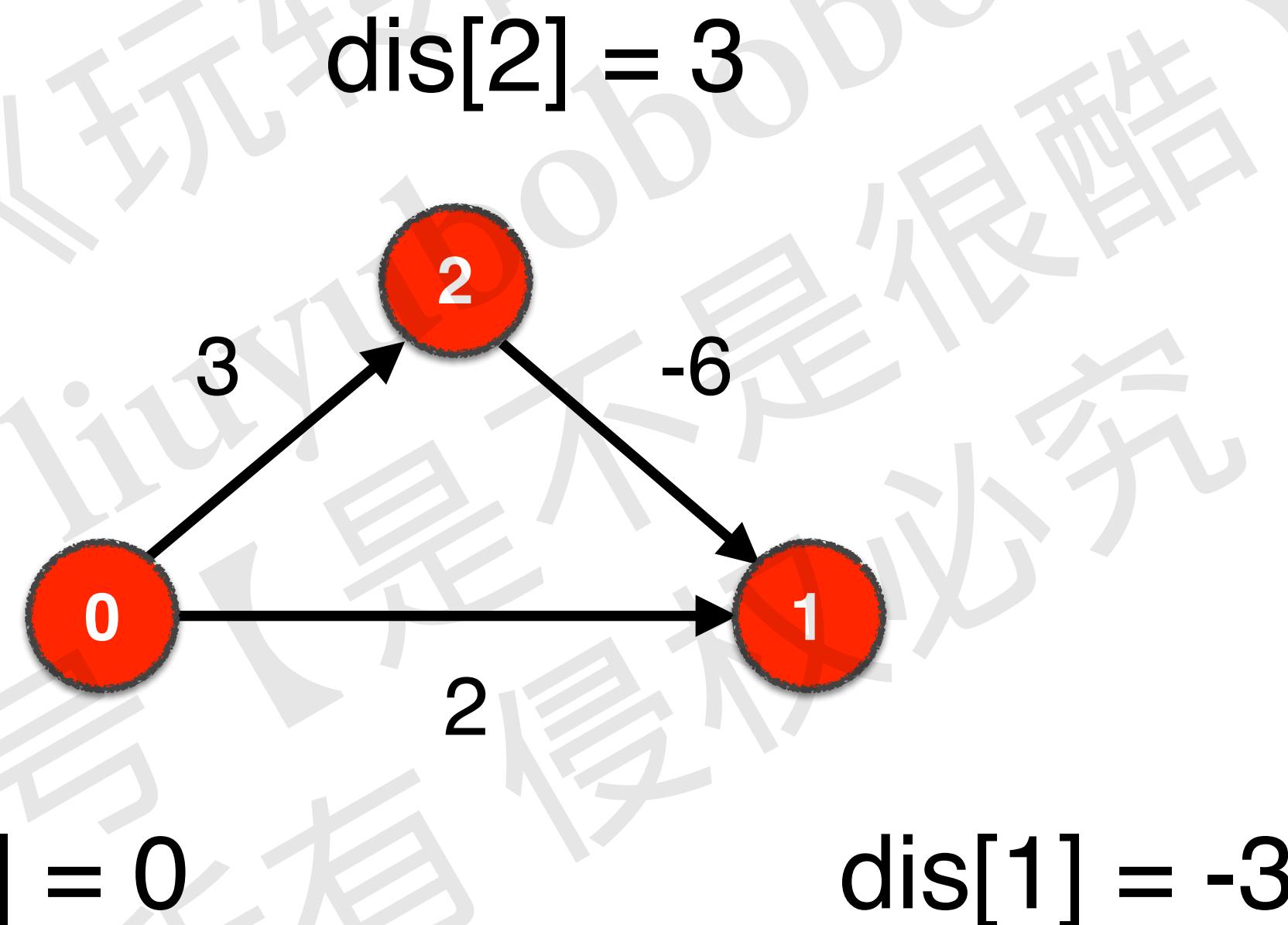
relax(2, 1)

relax(0, 1)

relax(0, 2)

relax(2, 1)

Bellman-Ford 算法



relax(2, 1)

relax(0, 1)

relax(0, 2)

relax(2, 1)

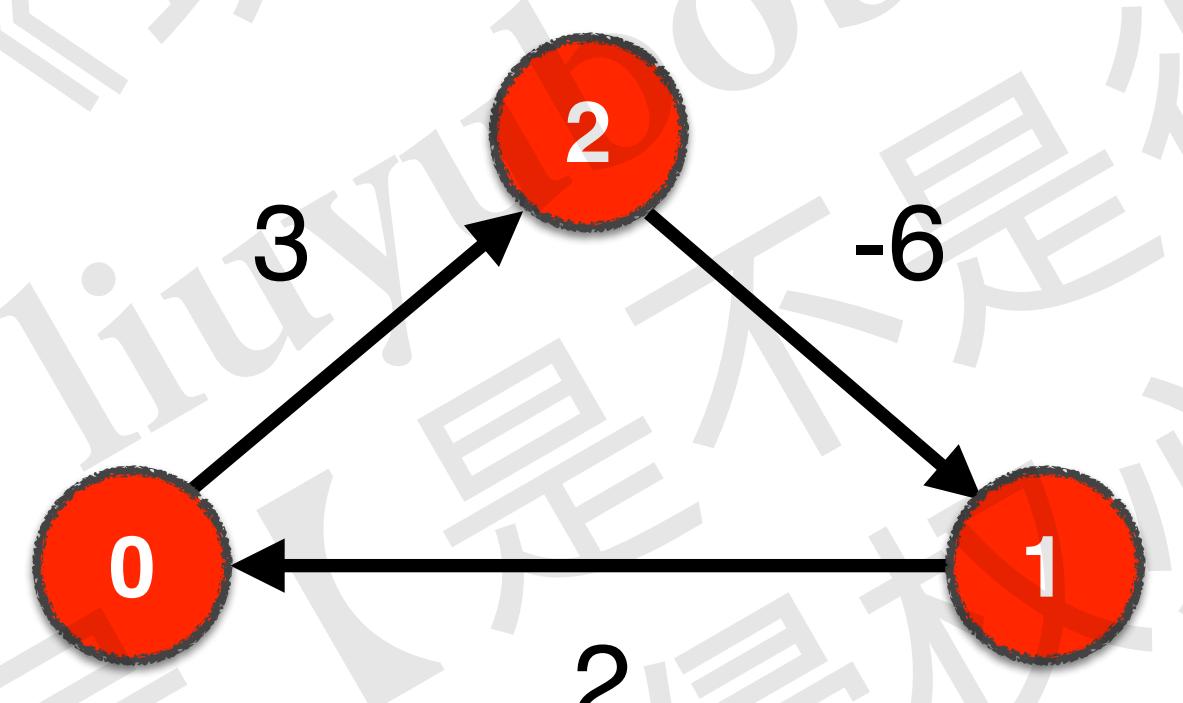
relax(0, 1)

relax(0, 2)

模拟3

慕课网《玩转图论算法》
讲师：liuyubobobo
公众账号：模拟人是很酷
版权所有 © 侵权必究

Bellman-Ford 算法



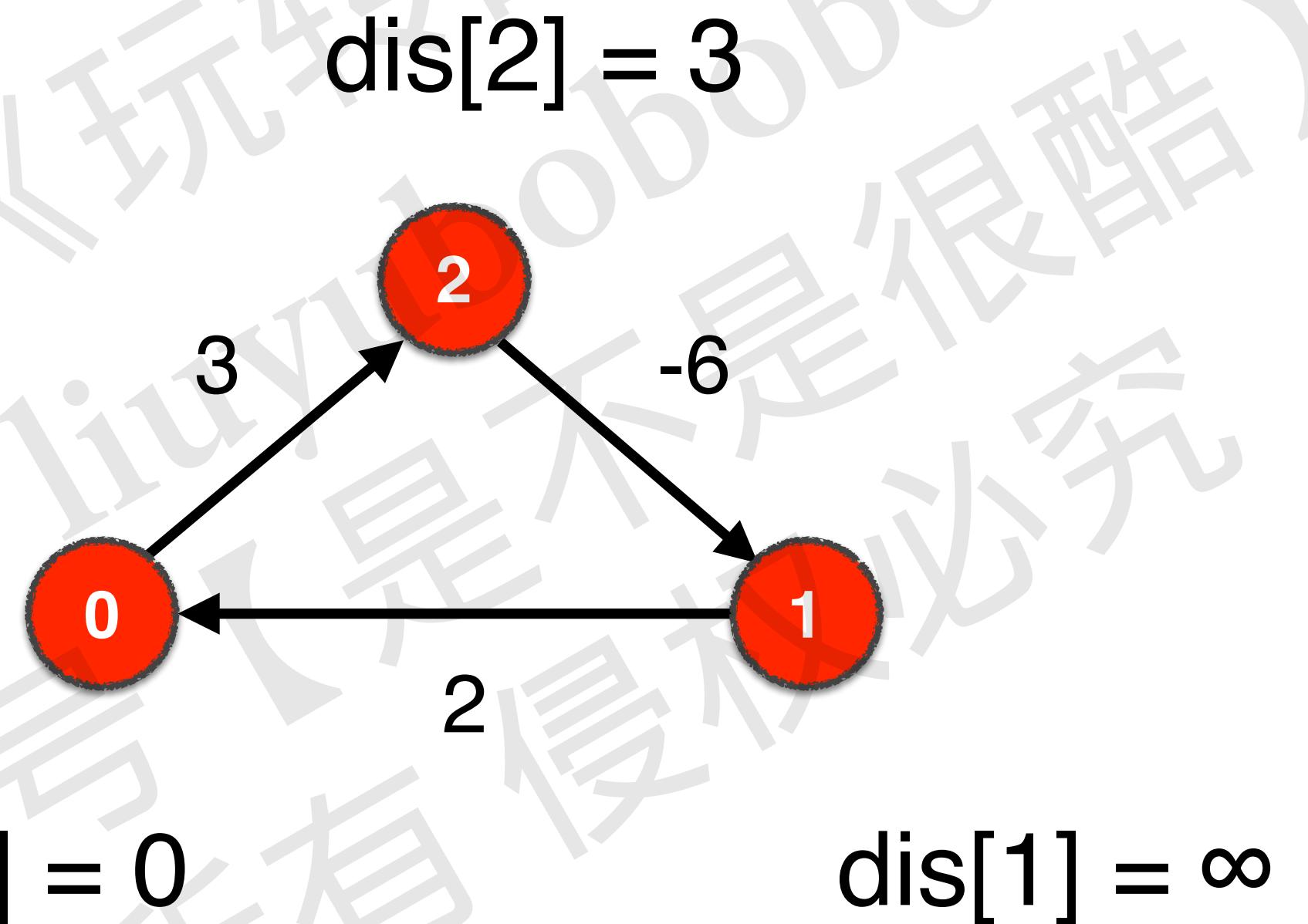
$dis[2] = \infty$

$relax(0, 2)$

$dis[0] = 0$

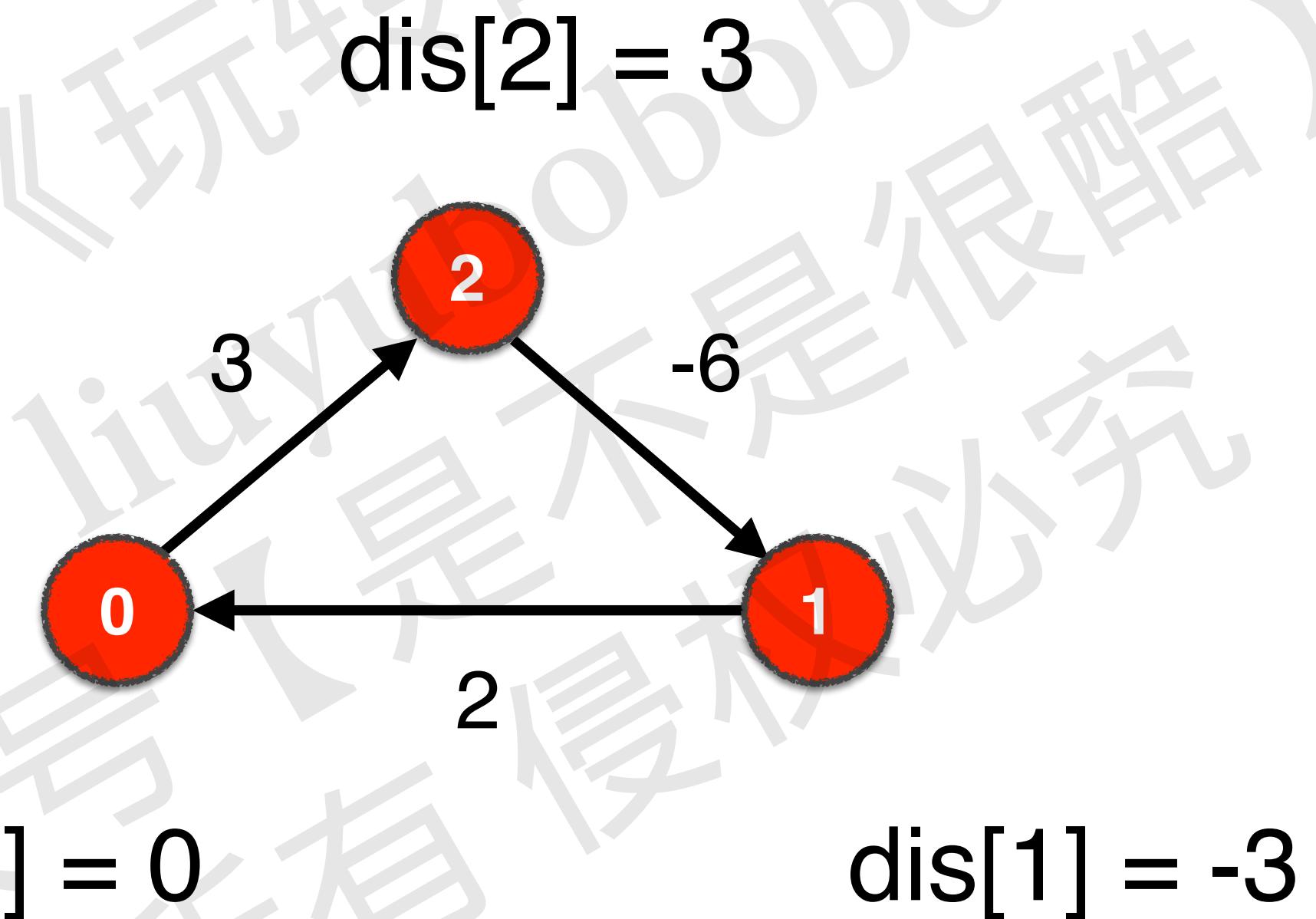
$dis[1] = \infty$

Bellman-Ford 算法



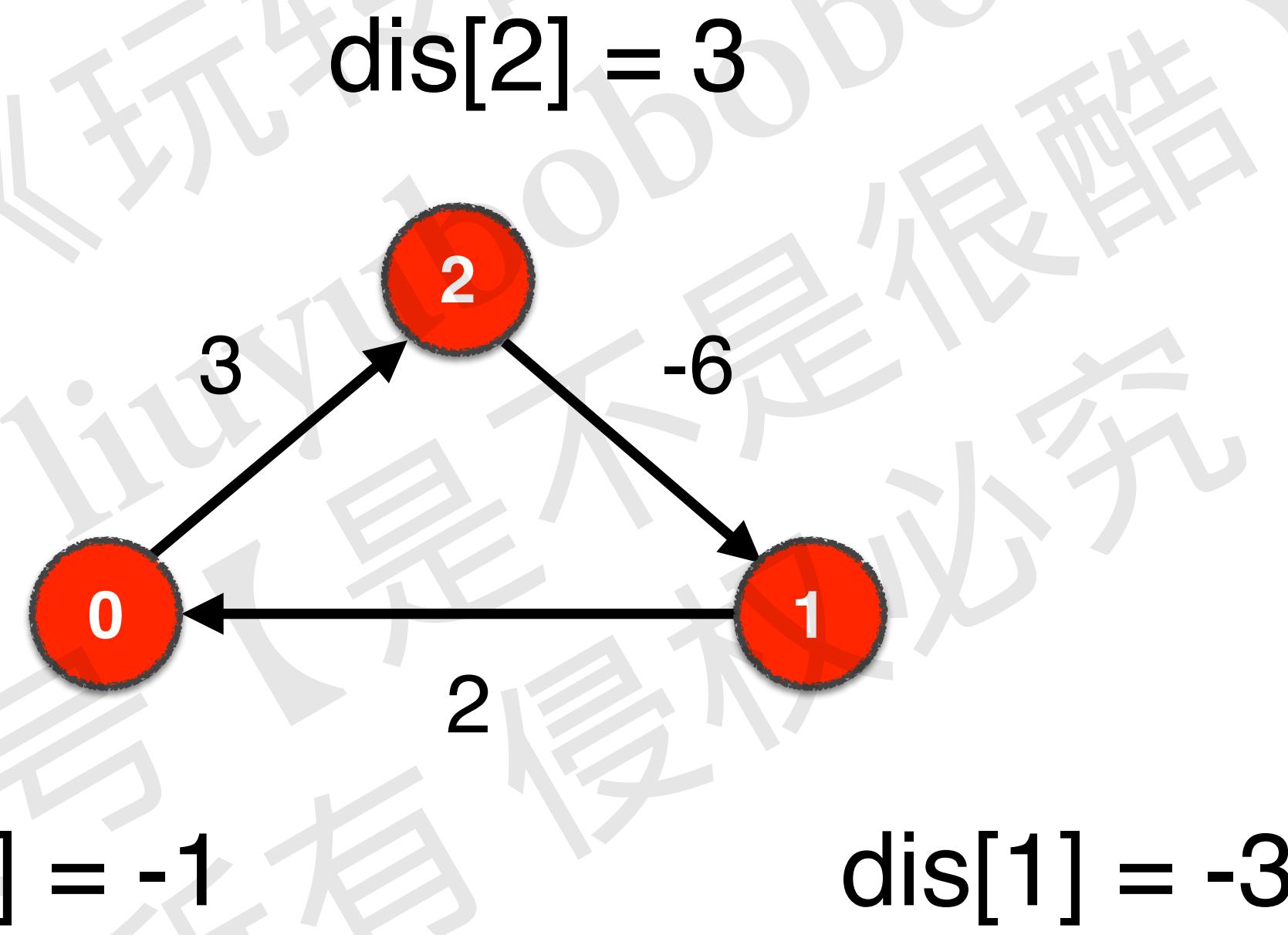
relax(0, 2)
relax(1, 0)
relax(2, 1)

Bellman-Ford 算法



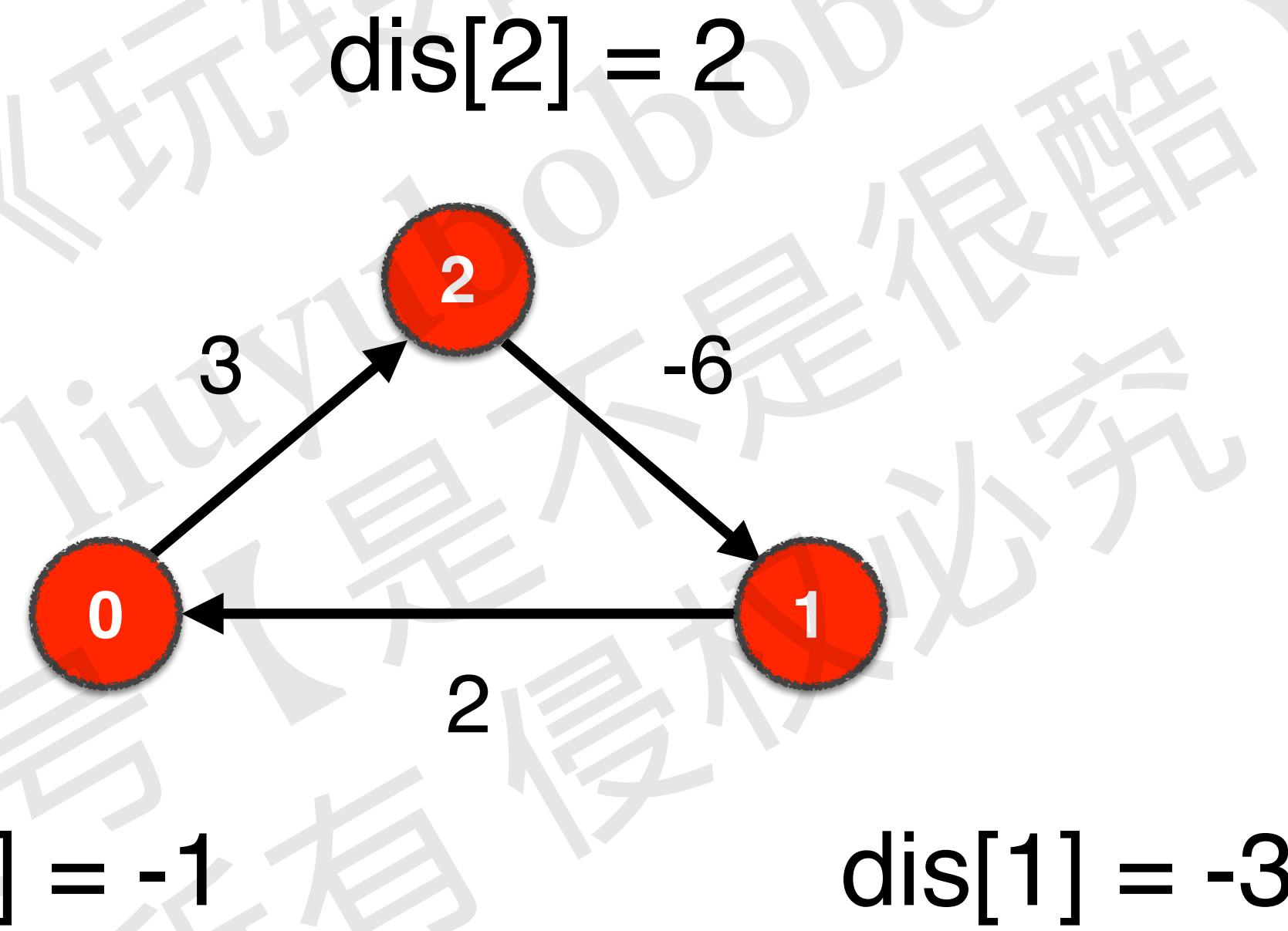
relax(0, 2)
relax(1, 0)
relax(2, 1)
relax(0, 2)
relax(1, 0)

Bellman-Ford 算法



relax(0, 2)
relax(1, 0)
relax(2, 1)
relax(0, 2)
relax(1, 0)
relax(2, 1)
relax(0, 2)

Bellman-Ford 算法



relax(0, 2)

relax(1, 0)

relax(2, 1)

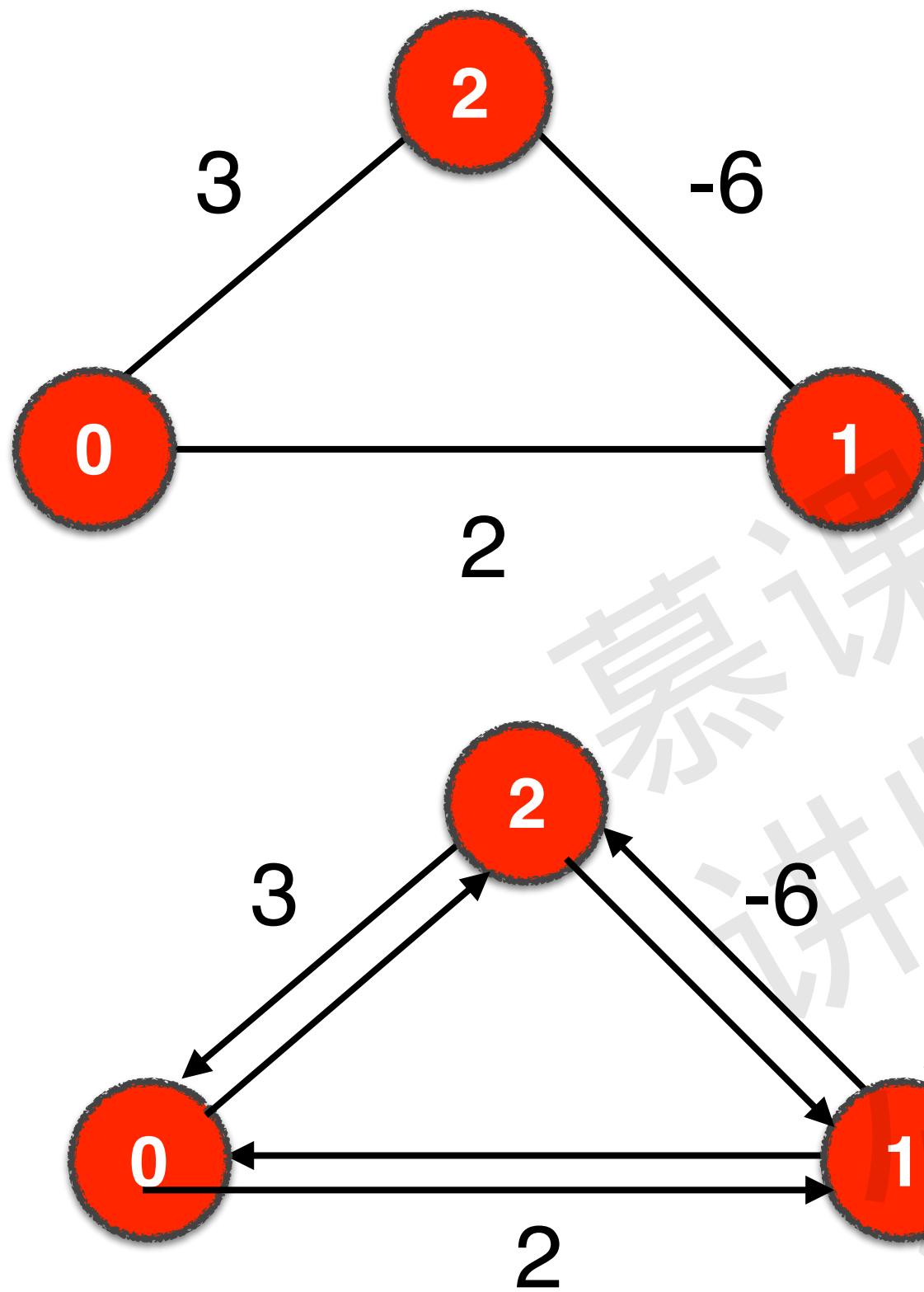
relax(0, 2)

relax(1, 0)

relax(2, 1)

relax(0, 2)

Bellman-Ford 算法



Bellman-Ford 算法：有向图算法

不包含负权环的最短路，
经过的边数最多为： $V-1$

找到经过的边数最多为 $V-1$ 的最短路

如果还存在边数为 V 的更短路，存在负权环

Bellman-Ford 算法

Bellman-Ford 算法：

初始 $dis[s] = 0$, 其余 dis 值为 ∞

对所有边进行一次松弛操作，则求出了到所有点，经过的边数最多为1的最短路

对所有边再进行一次松弛操作，则求出了到所有点，经过的边数最多为2的最短路

...

对所有边进行 $V - 1$ 次松弛操作，则求出了到所有点，经过的边数最多为 $V - 1$ 的最短路

如果对所有边再进行一次松弛操作，还能更新 dis ，则说明图中包含负权环

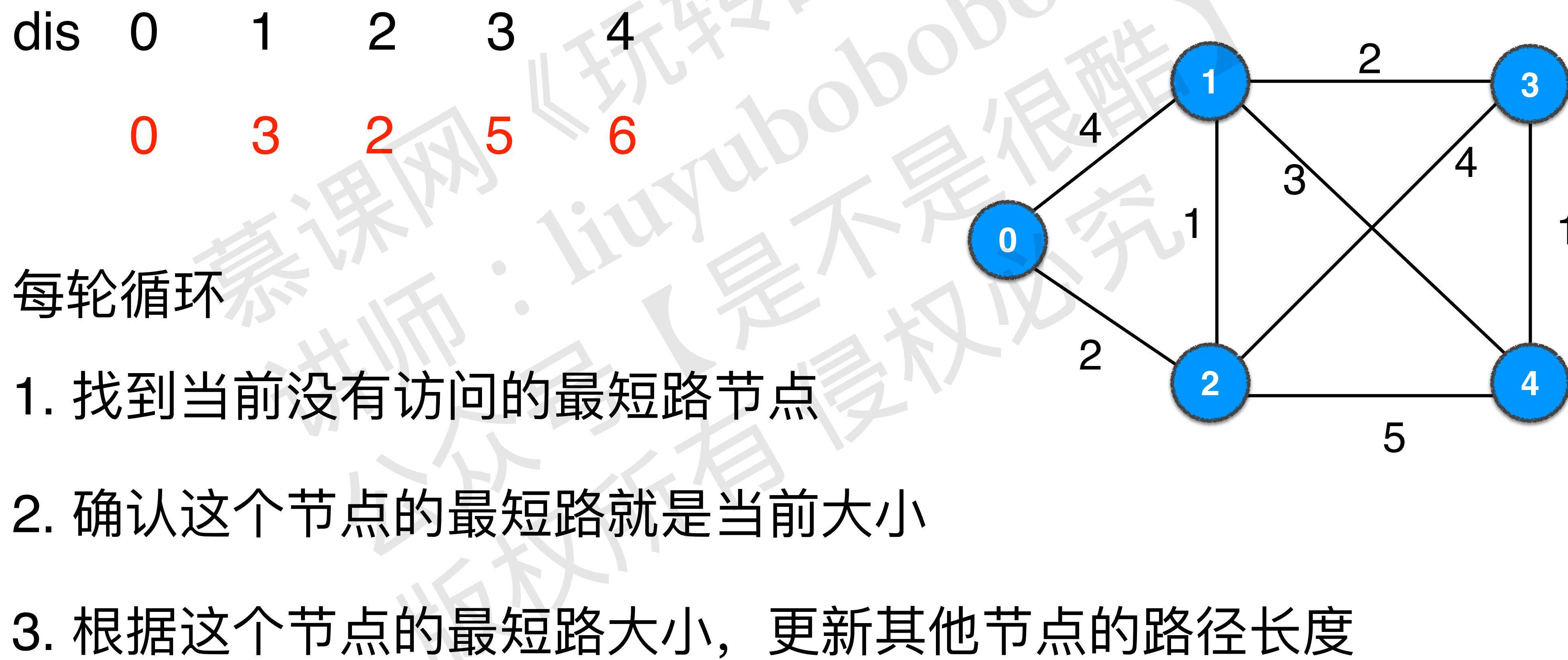
实现 Bellman-Ford 算法

liuyubobobo

编程实践：实现 Bellman-Ford 算法

慕课网 · liuyanbo · bobo 《玩转图论算法》

Dijkstra 算法



Bellman-Ford 算法

Bellman-Ford 算法: $O(V^*E)$

初始 $dis[s] = 0$, 其余 dis 值为 ∞

对所有边进行一次松弛操作, 则求出了到所有点, 经过的边数最多为1的最短路

对所有边再进行一次松弛操作, 则求出了到所有点, 经过的边数最多为2的最短路

...

对所有边进行 $V - 1$ 次松弛操作, 则求出了到所有点, 经过的边数最多为 $V - 1$ 的最短路

如果对所有边再进行一次松弛操作, 还能更新 dis , 则说明图中包含负权环

更多关于 Bellman-Ford 算法的讨论

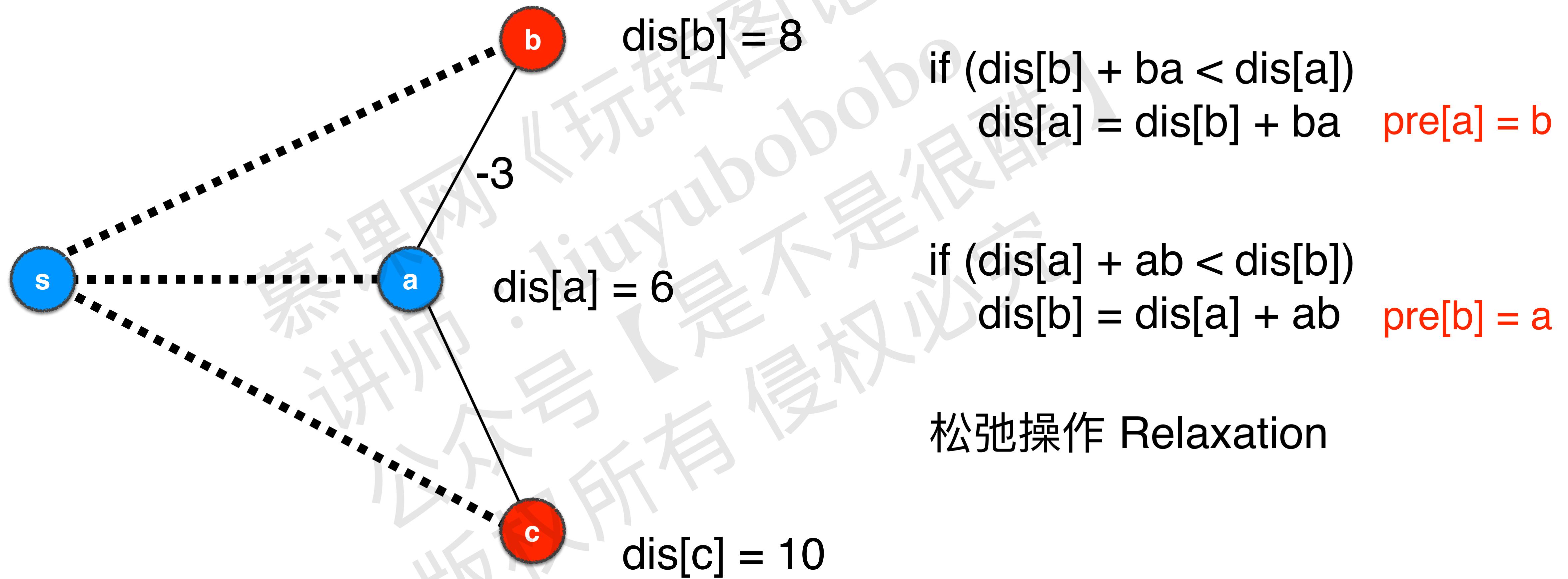
liuyubobobo

更多关于 Bellman-Ford 算法的讨论

求解具体路径?

使用 pre 数组

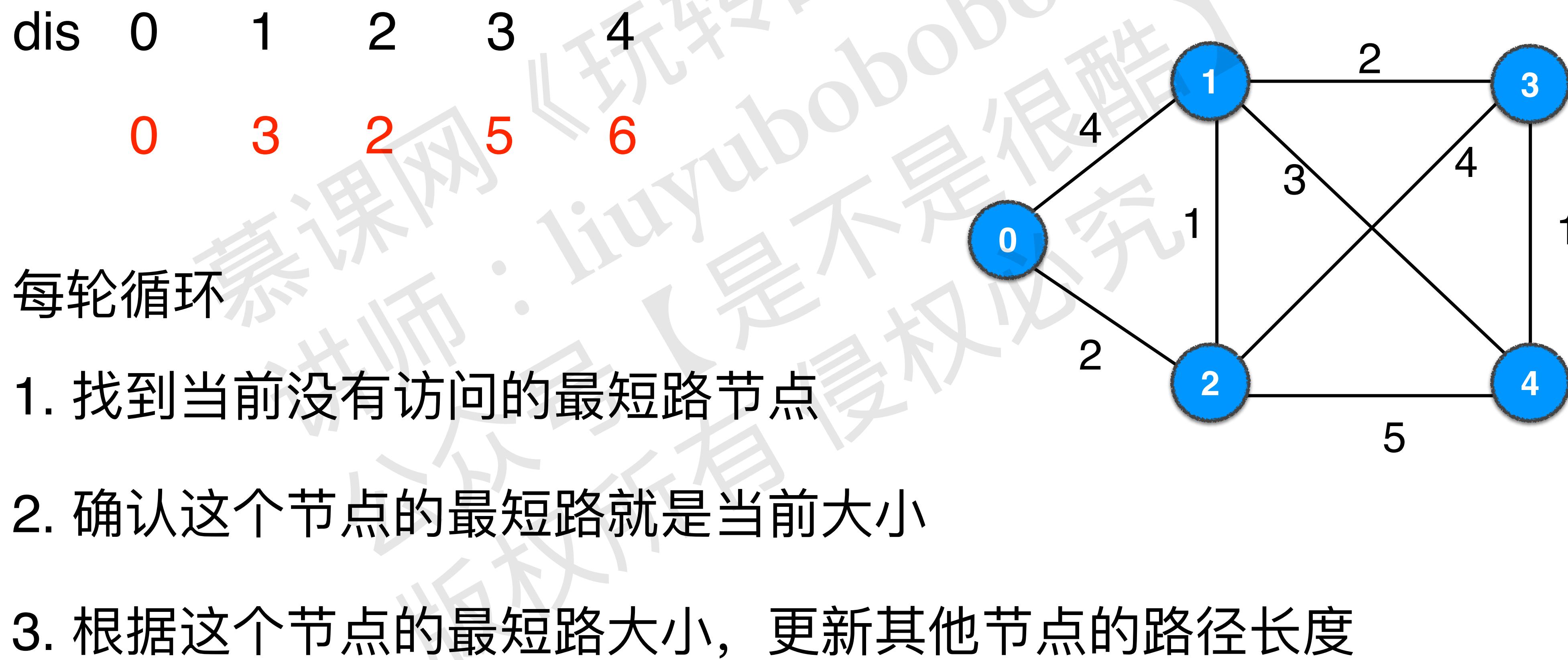
Bellman-Ford 算法



编程实践：Bellman-Ford 求解具体路径

慕课网 · liuyubobobo · 《玩转图论算法》

Dijkstra 算法



更多关于 Bellman-Ford 算法的讨论

只关注从 s 到 t 之间的最短路径? 不能提前终止

Bellman-Ford 算法优化 SPFA (Shortest Path Fast Algorithm)

西南交通大学: 段凡丁 (1994)

Edward F. Moore (1959)

最差复杂度: $O(VE)$

Floyd 算法

liuyubobobo

Floyd 算法

所有点对最短路径

图的直径

所有点对最短路径中的最大值

Floyd 算法

所有点对最短路径

Dijkstra: $O(VE \log E)$

不能包含负权边

Bellman-Ford: $O(V^*V^*E)$

可以包含负权边；检测负权环

Floyd: $O(V^*V^*V)$

可以包含负权边；检测负权环

Floyed 算法

Floyed: $O(V^*V^*V)$

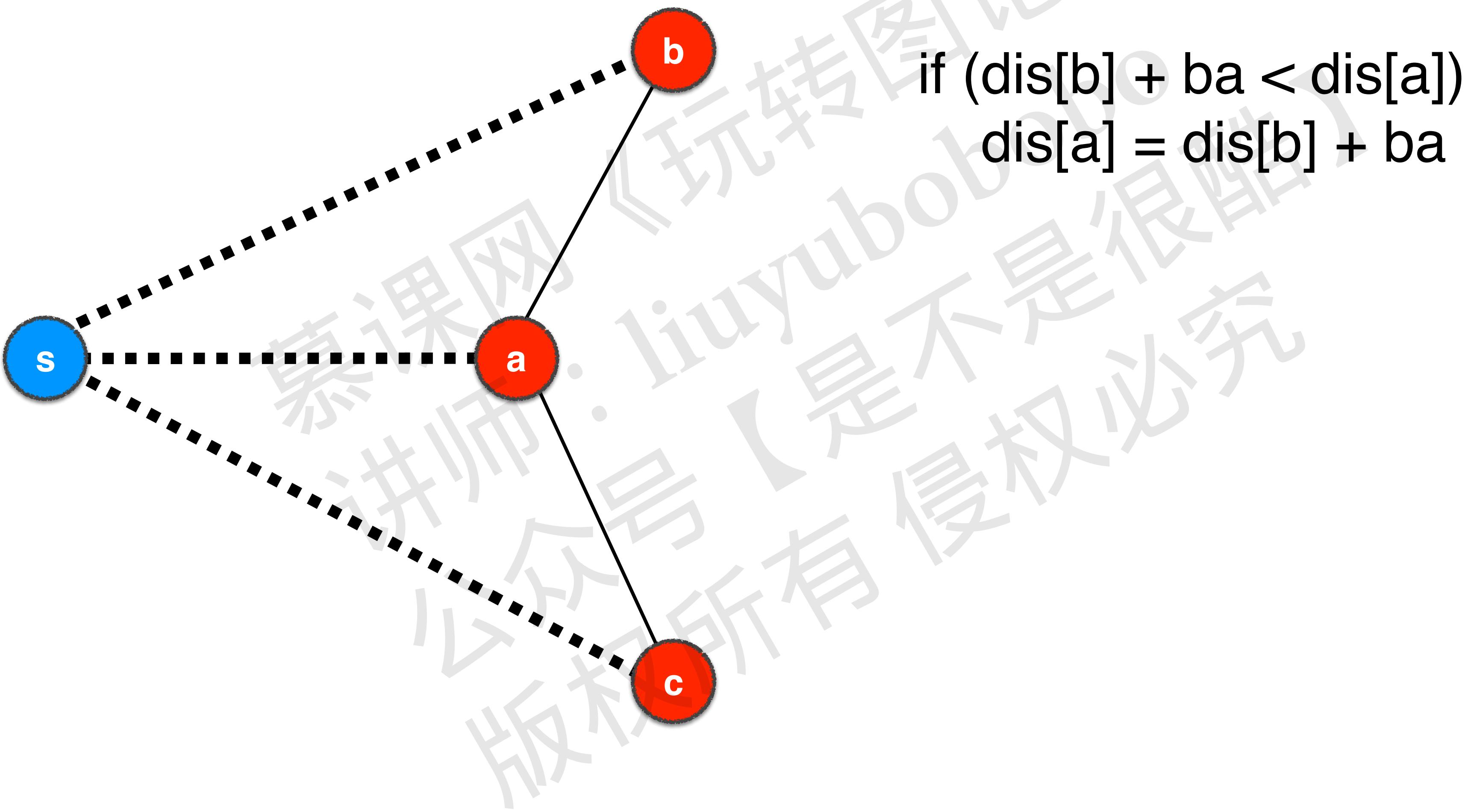
可以包含负权边；检测负权环

Floyed-Warshall

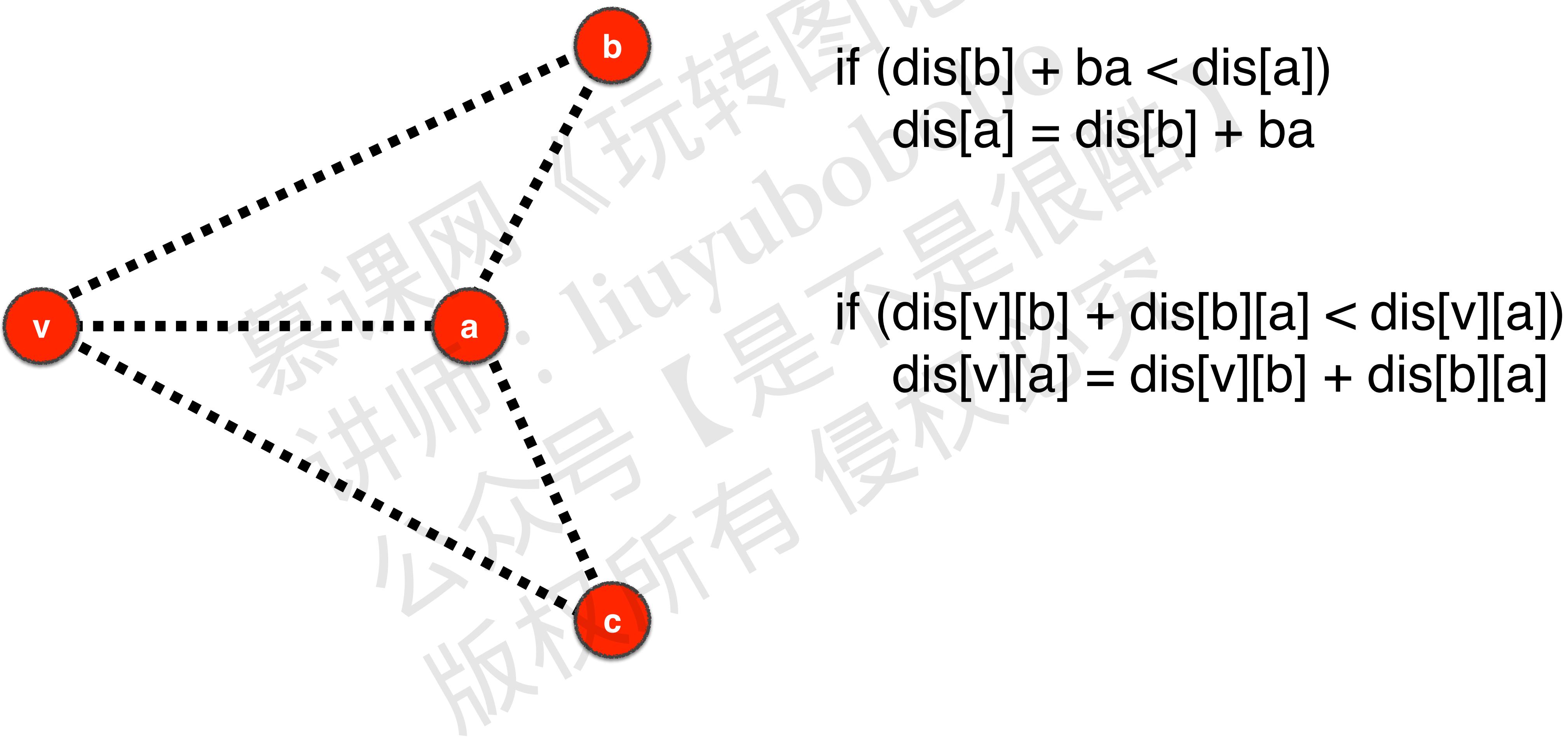
Roy-Floyed

Roy-Warshall

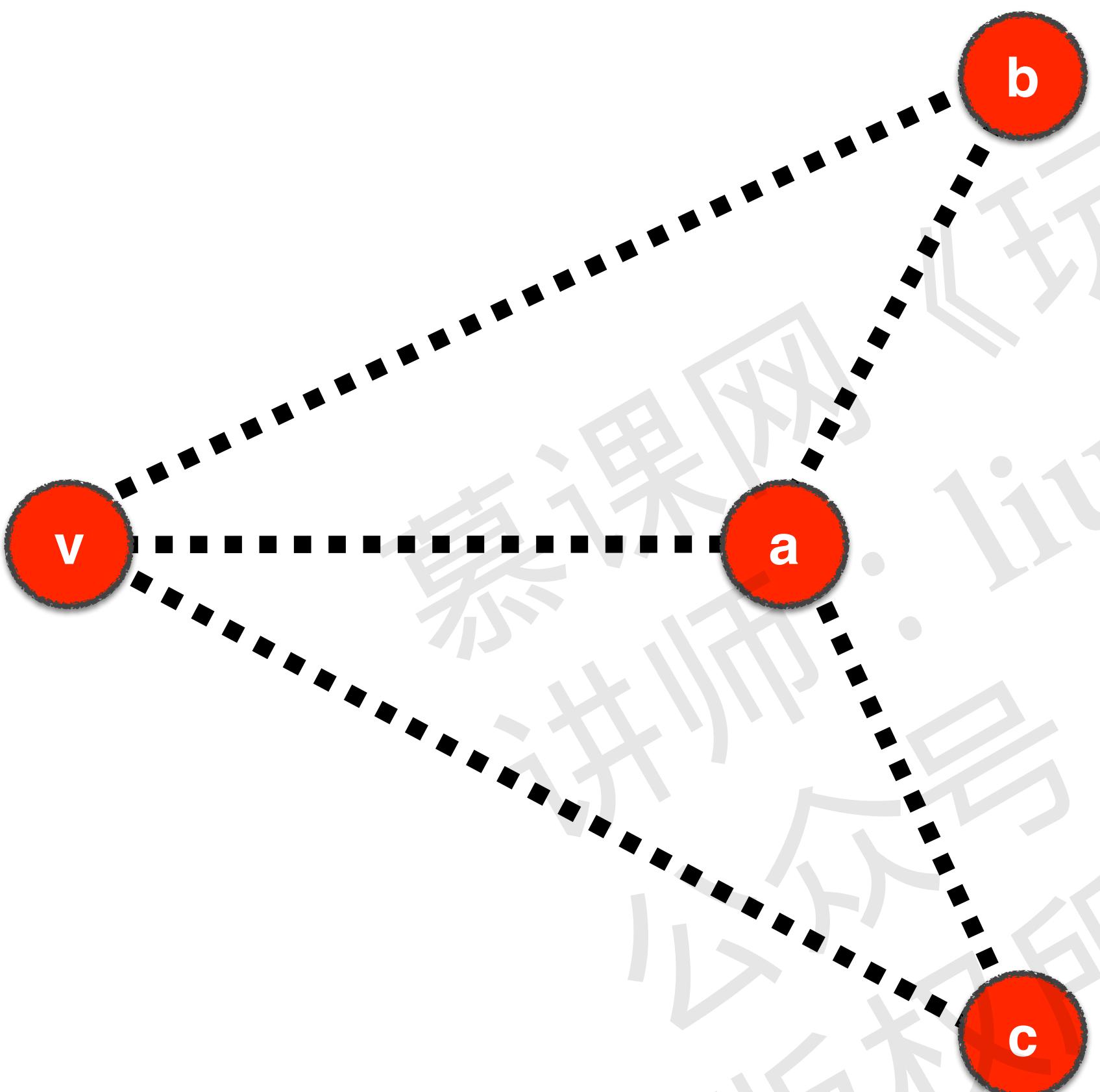
Floyd 算法



Floyed 算法



Floyd 算法



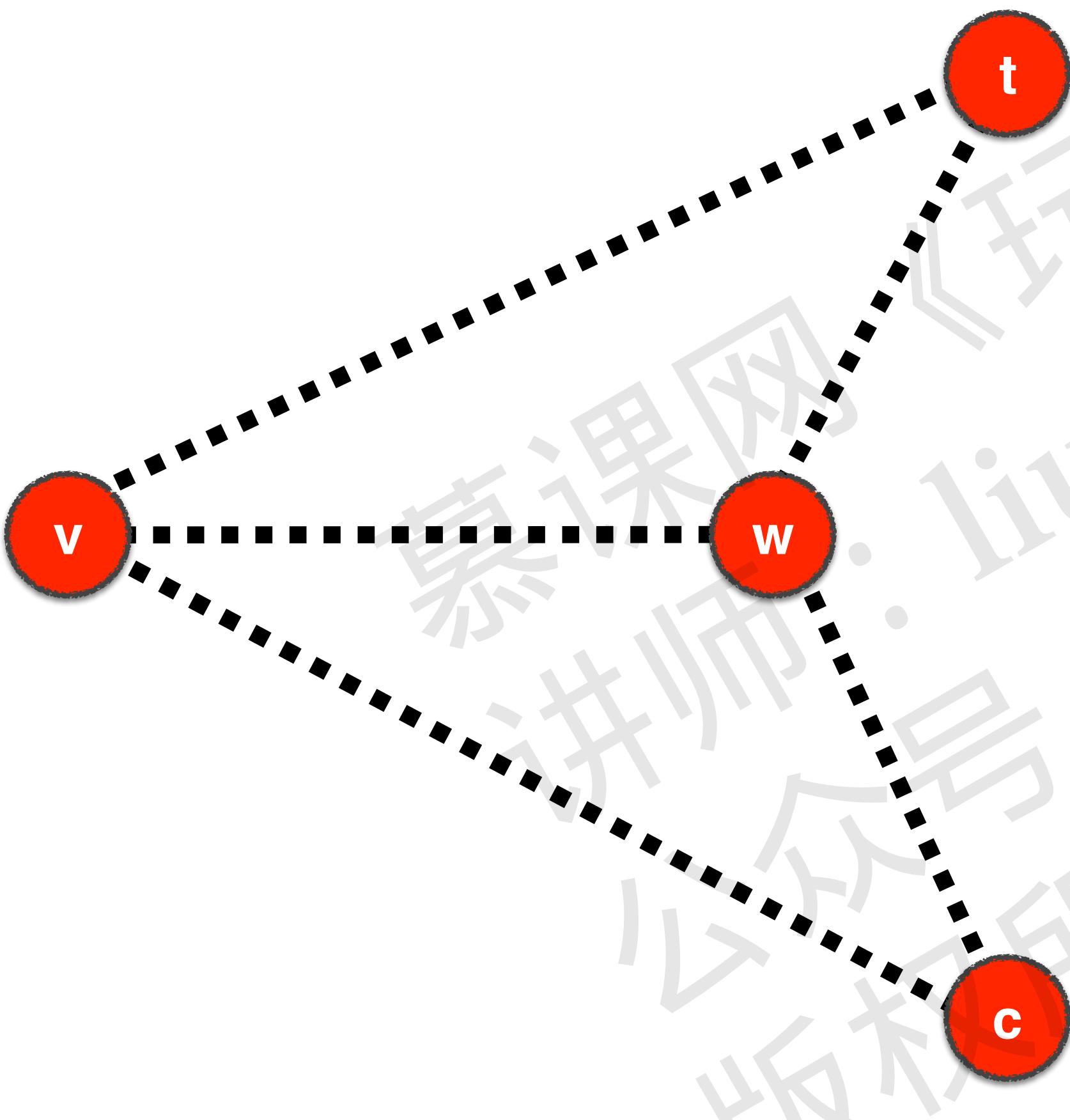
```
if (dis[v][b] + dis[b][a] < dis[v][a])  
    dis[v][a] = dis[v][b] + dis[b][a]
```

初始，如果v-w有边， $dis[v][w]=vw$ ； $dis[v][v]=0$

否则， $dis[v][w] = \infty$

```
for(int t = 0; t < V; t ++)  
    for(int v = 0; v < V; v ++)  
        for(int w = 0; w < V; w ++)  
            if(dis[v][t] + dis[t][w] < dis[v][w])  
                dis[v][w] = dis[v][t] + dis[t][w]
```

Floyd 算法



初始, 如果 $v-w$ 有边, $dis[v][w]=vw$; $dis[v][v]=0$

否则, $dis[v][w] = \infty$

```
for(int t = 0; t < V; t ++)
```

```
    for(int v = 0; v < V; v ++)
```

```
        for(int w = 0; w < V; w ++)
```

```
            if(dis[v][t] + dis[t][w] < dis[v][w])
```

```
                dis[v][w] = dis[v][t] + dis[t][w]
```

Floyd 算法

初始，如果 $v-w$ 有边， $\text{dis}[v][w]=vw$ ； $\text{dis}[v][v]=0$

否则， $\text{dis}[v][w] = \infty$

```
for(int t = 0; t < V; t ++)  
    for(int v = 0; v < V; v ++)  
        for(int w = 0; w < V; w ++)  
            if(dis[v][t] + dis[t][w] < dis[v][w])  
                dis[v][w] = dis[v][t] + dis[t][w]
```

每轮循环求解出中间经过 [0...t]
这些点的最短路径

负权环?  $\text{dis}[v][v] < 0?$

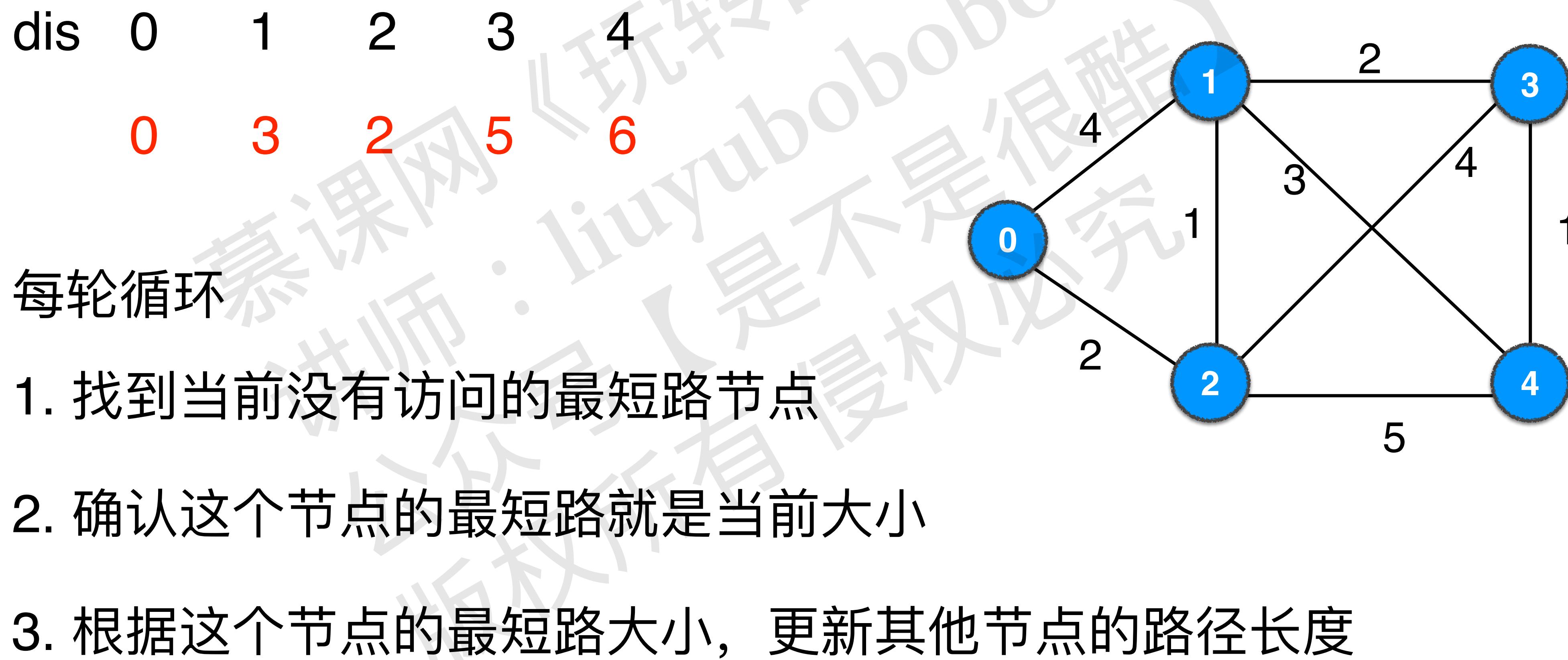
实现 Floyd 算法

liuyubobobo

编程实践：实现 Floyed 算法

慕课网 · liuyibobobo · 《玩转图论算法》

Dijkstra 算法



本章小结和 更多关于最短路径的讨论

liuyubobobo

本章小结

	单源最短路径	所有点对最短路径
Dijkstra	$O(E \log E)$	
	$O(E \log V)$	
	不能包含负权边	
Bellman-Ford	$O(V^* E)$ SPFA	Floyd $O(V^3)$
	可以包含负权边；	可以包含负权边；
	检测负权环	检测负权环

更多关于最短路径的讨论

动态规划

Dijkstra

$dis[v]$

贪心

Bellman-Ford

$dis[v]$

Floyed

$dis[v][w]$

更多关于最短路径的讨论

最长路径?

指数算法

欧拉路径 vs 哈密尔顿路径

最短路径 vs 最长路径

大家加油！

欢迎大家关注我的个人公众号：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

「是不是很酷」

玩儿转图论算法

liuyubobo