

玩儿转图论算法

liuyubobobo

欧拉回路和欧拉路径

liuyubobobo

欧拉回路

liuyubobobo

慕课网 《玩转图论算法》
讲师：liuyubobobo
公众账号：达乐极客
版权所有 © liuyubobobo

欧拉回路

通常面试不涉及

慕课网《玩转图论算法》
讲师：liuyubobobo
公众号【这下是很酷】
版权所有，侵权必究

欧拉回路

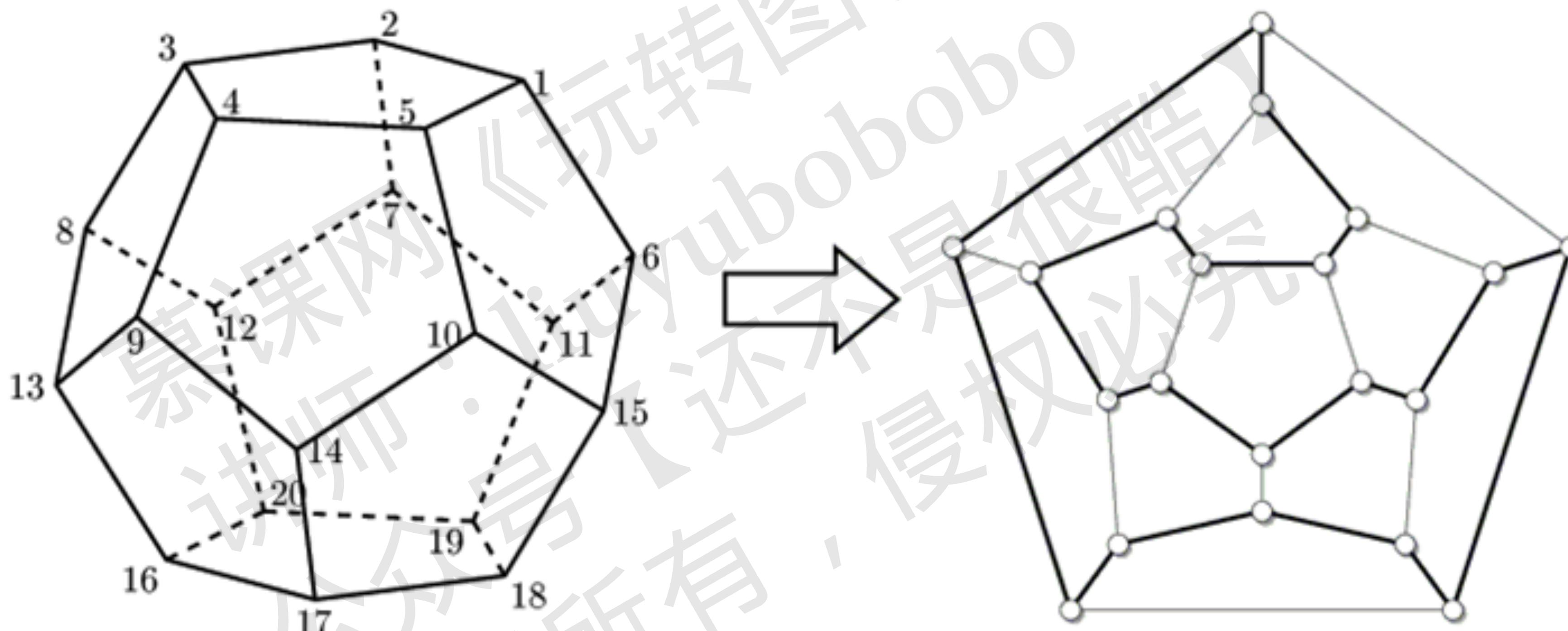
哈密尔顿回路

从一个点出发，沿着边行走，**经过每个顶点恰好一次**，之后再回到出发点

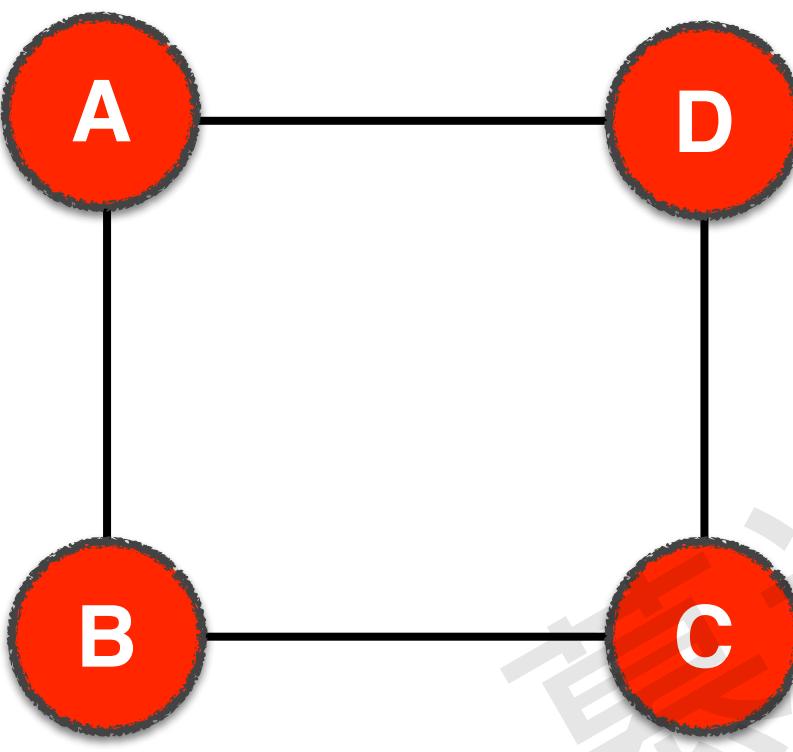
欧拉回路

从一个点出发，沿着边行走，**经过每个边恰好一次**，之后再回到出发点

欧拉回路

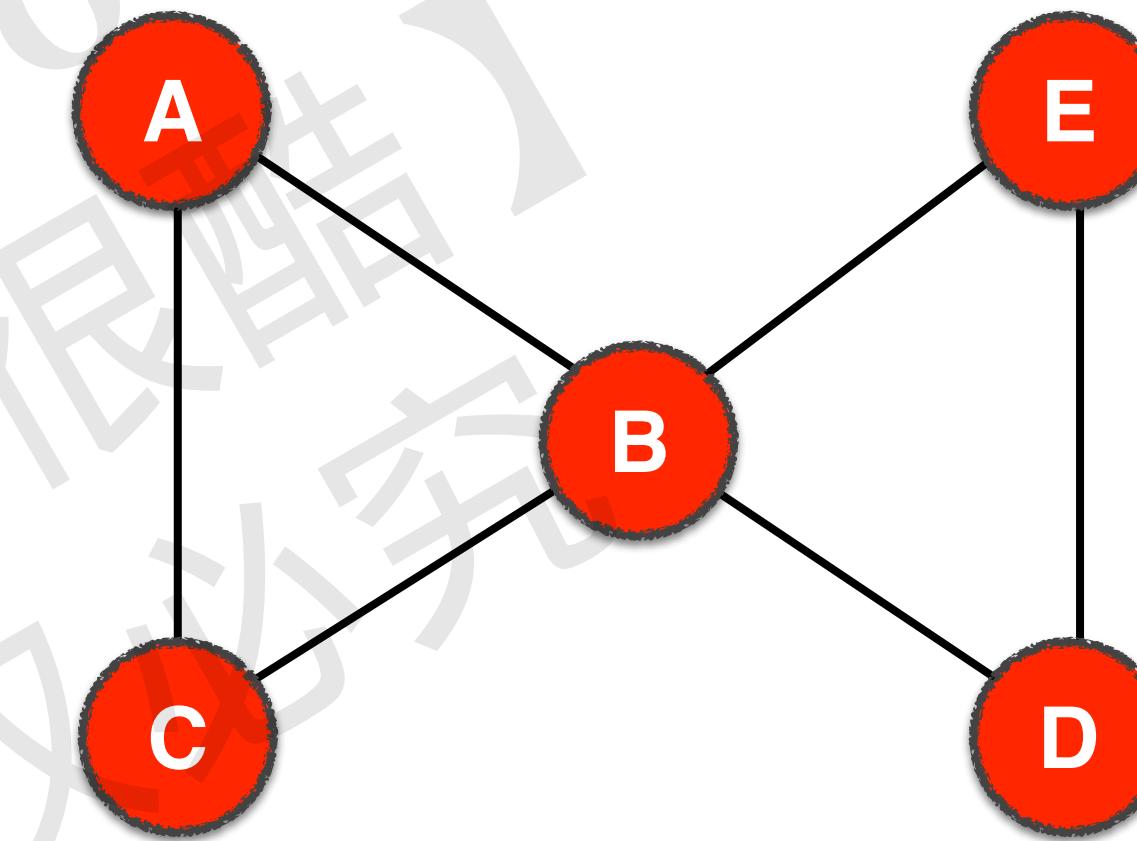


欧拉回路



A - B - C - D - A

既是哈密尔顿回路，也是欧拉回路

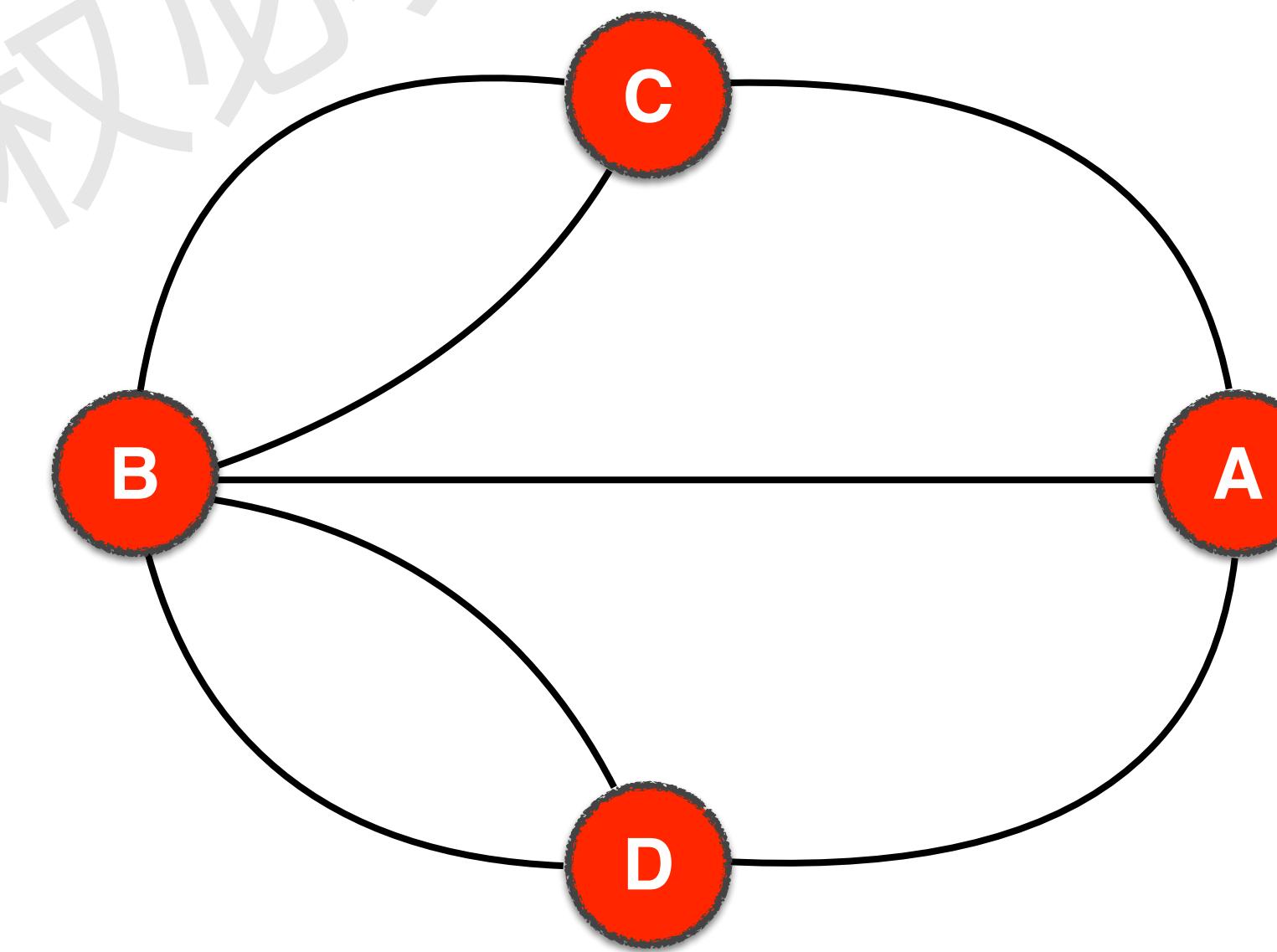
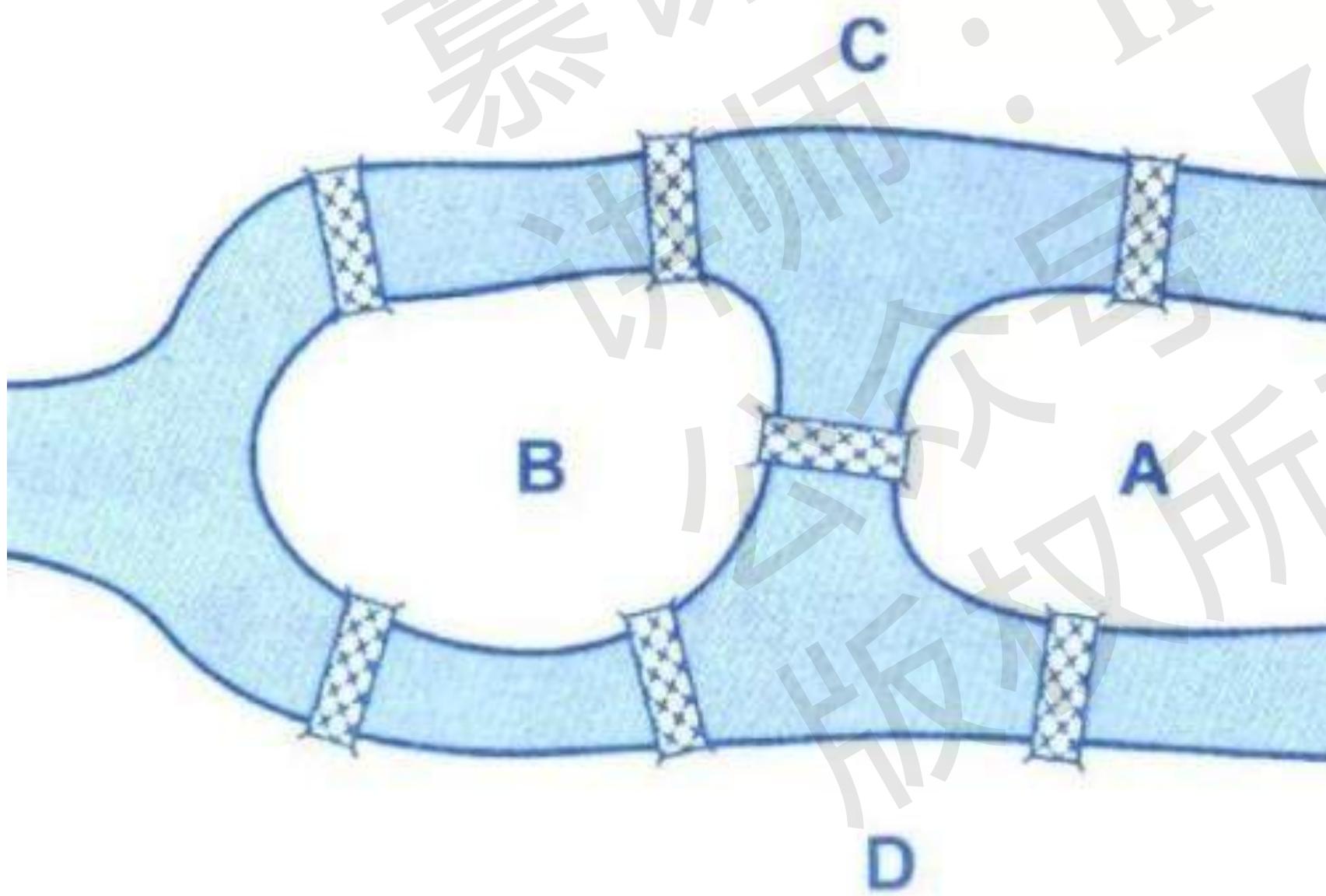


A - B - D - E - B - C - A

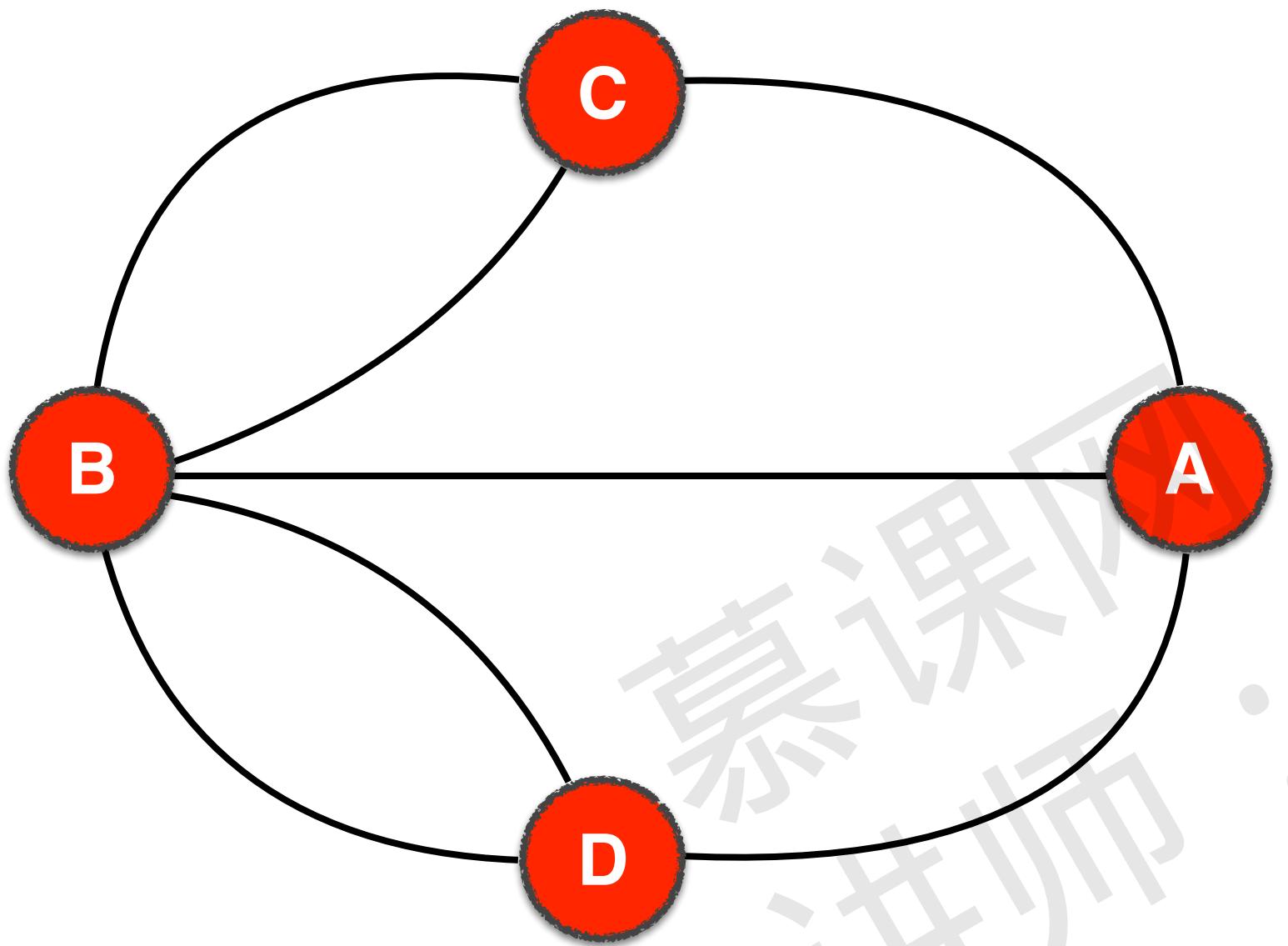
欧拉回路，没有哈密尔顿回路

欧拉回路

欧拉回路是数学家欧拉在研究著名的德国哥尼斯堡(Koenigsberg)七桥问题时发现的。如图所示，流经哥尼斯堡的普雷格尔河中有两个岛，两个岛与两岸共4处陆地通过7座桥彼此相联。7桥问题就是如何能从任一处陆地出发，经过且经过每个桥一次后回到原出发点。



欧拉回路



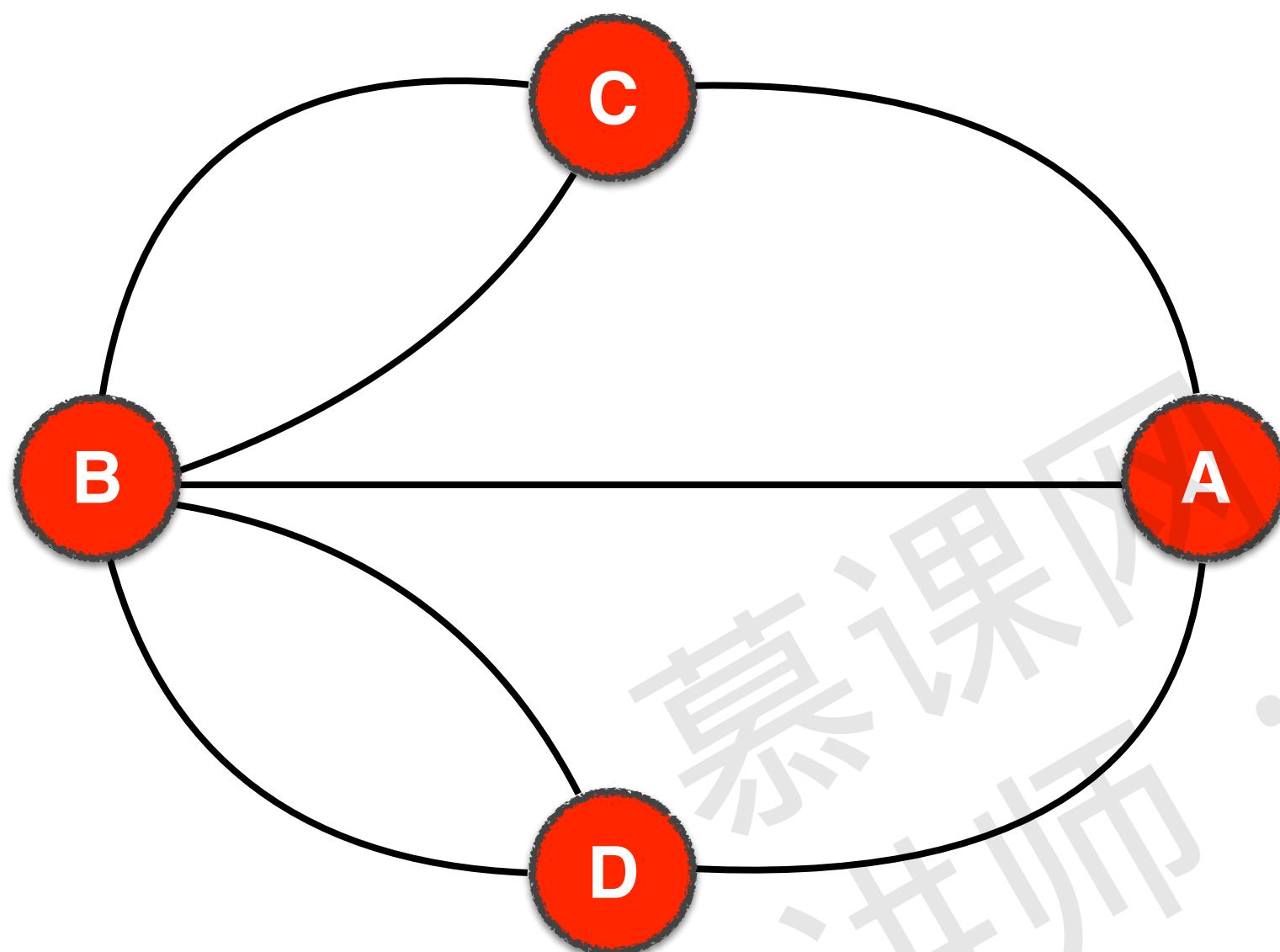
很长时间没有人能解决这个问题

欧拉解决问题的思路——证明这个问题没有解

欧拉回路存在的性质

liuyubobobo

欧拉回路存在的性质



很长时间没有人能解决这个问题

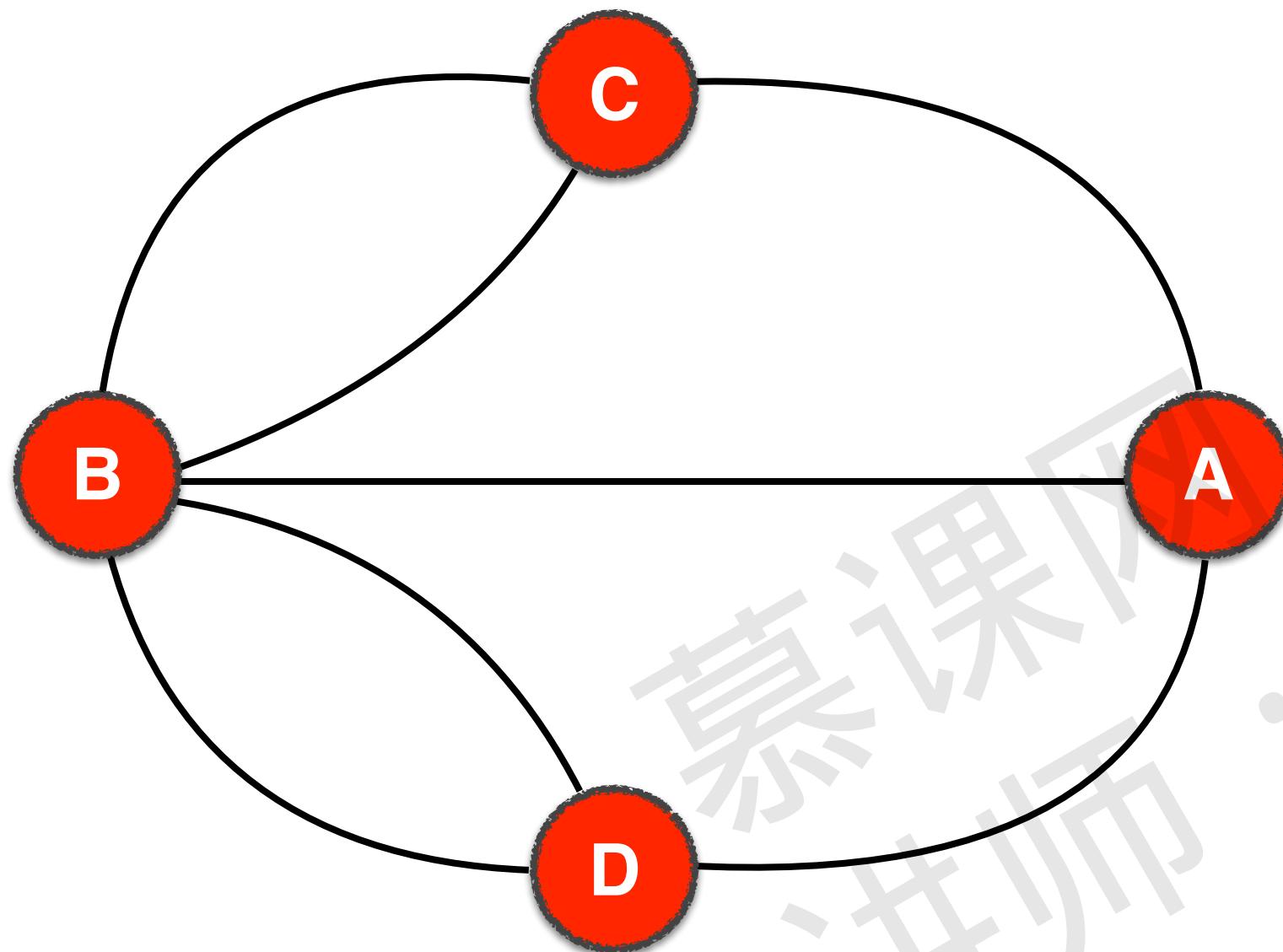
欧拉解决问题的思路——证明这个问题没有解

对于每一个点，进去一次，出去一次，“耗费两条边”

要想让每条边的都走一遍，回到原点，每个点都必须有进有出

每个点的相连边数（即每个点的度）必须是偶数

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数

图存在欧拉回路

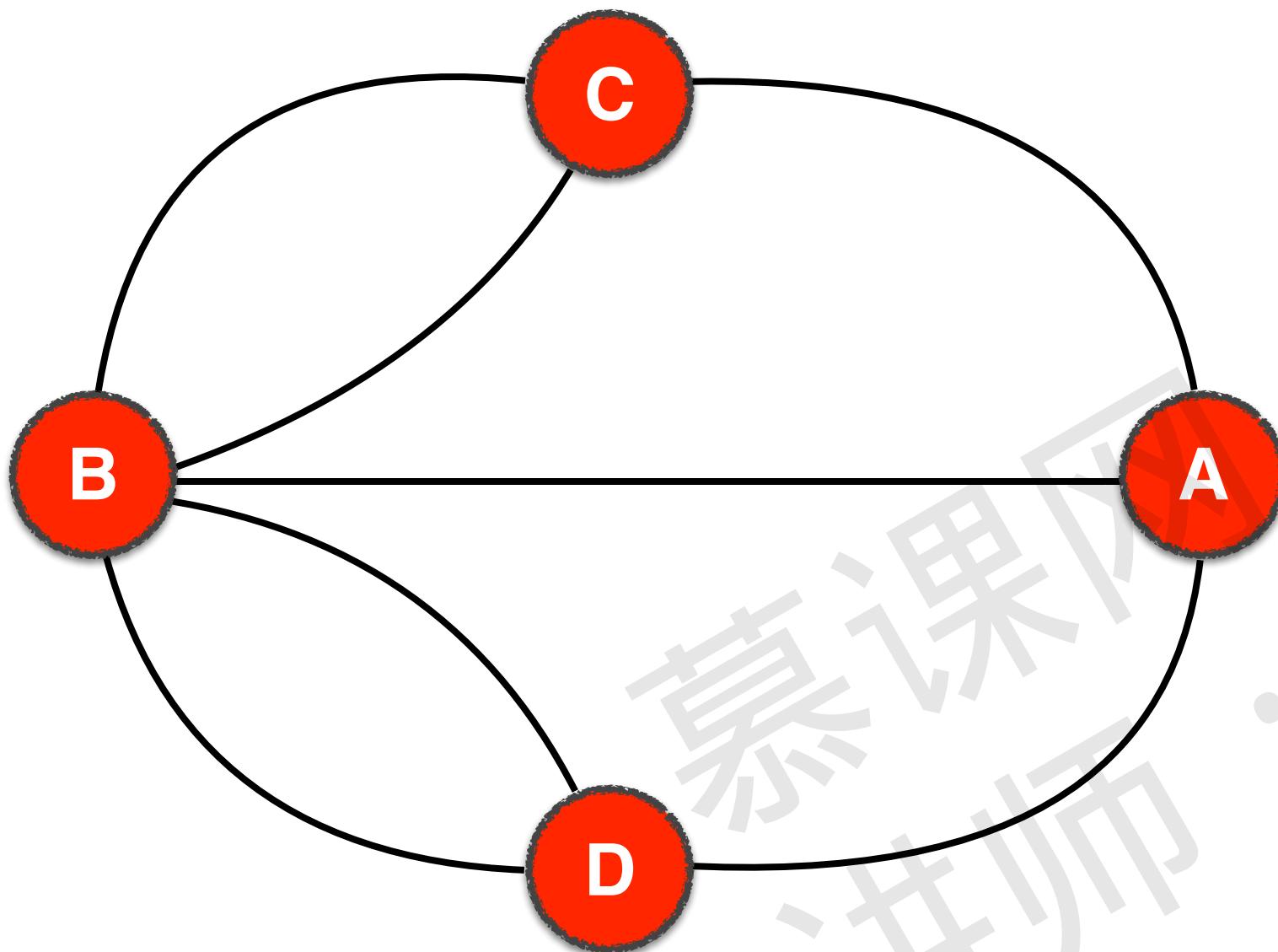
每个点的度是偶数

图存在欧拉回路

每个点的度是偶数

图存在欧拉回路

欧拉回路存在的性质

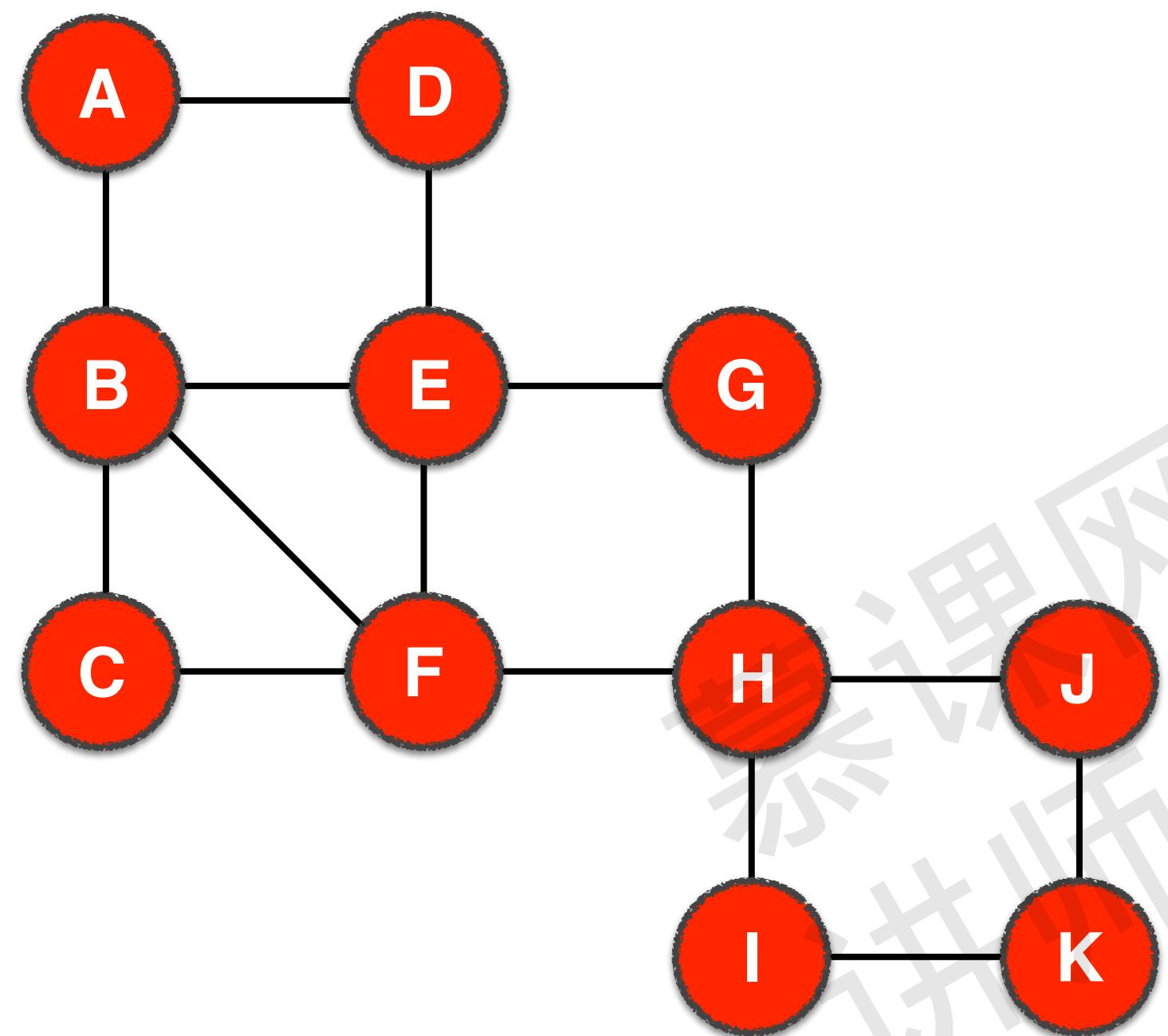


对于无向联通图
每个点的度是偶数



图存在欧拉回路

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



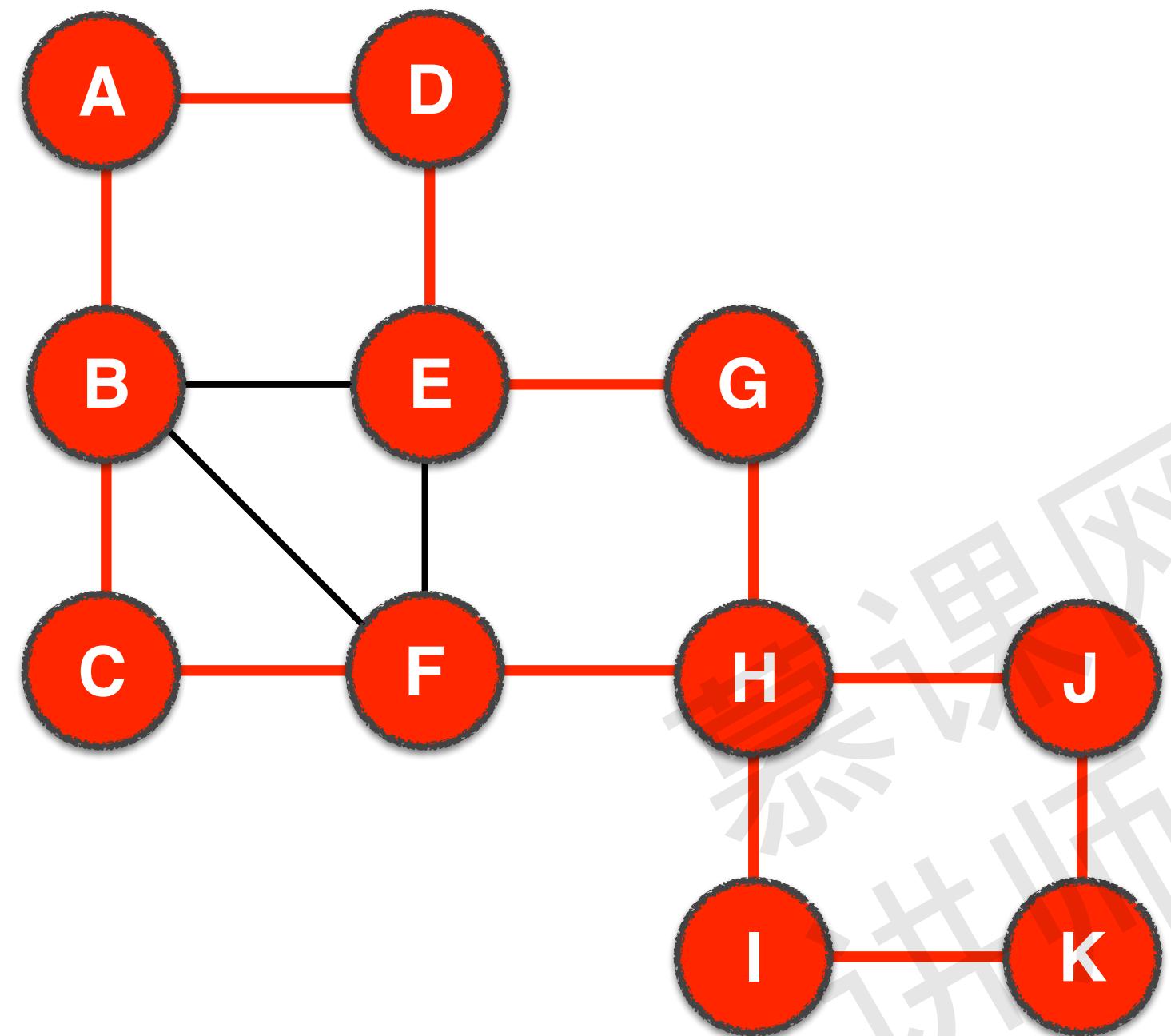
图存在欧拉回路

从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

欧拉回路存在的性质



对于无向联通图
每个点的度是偶数



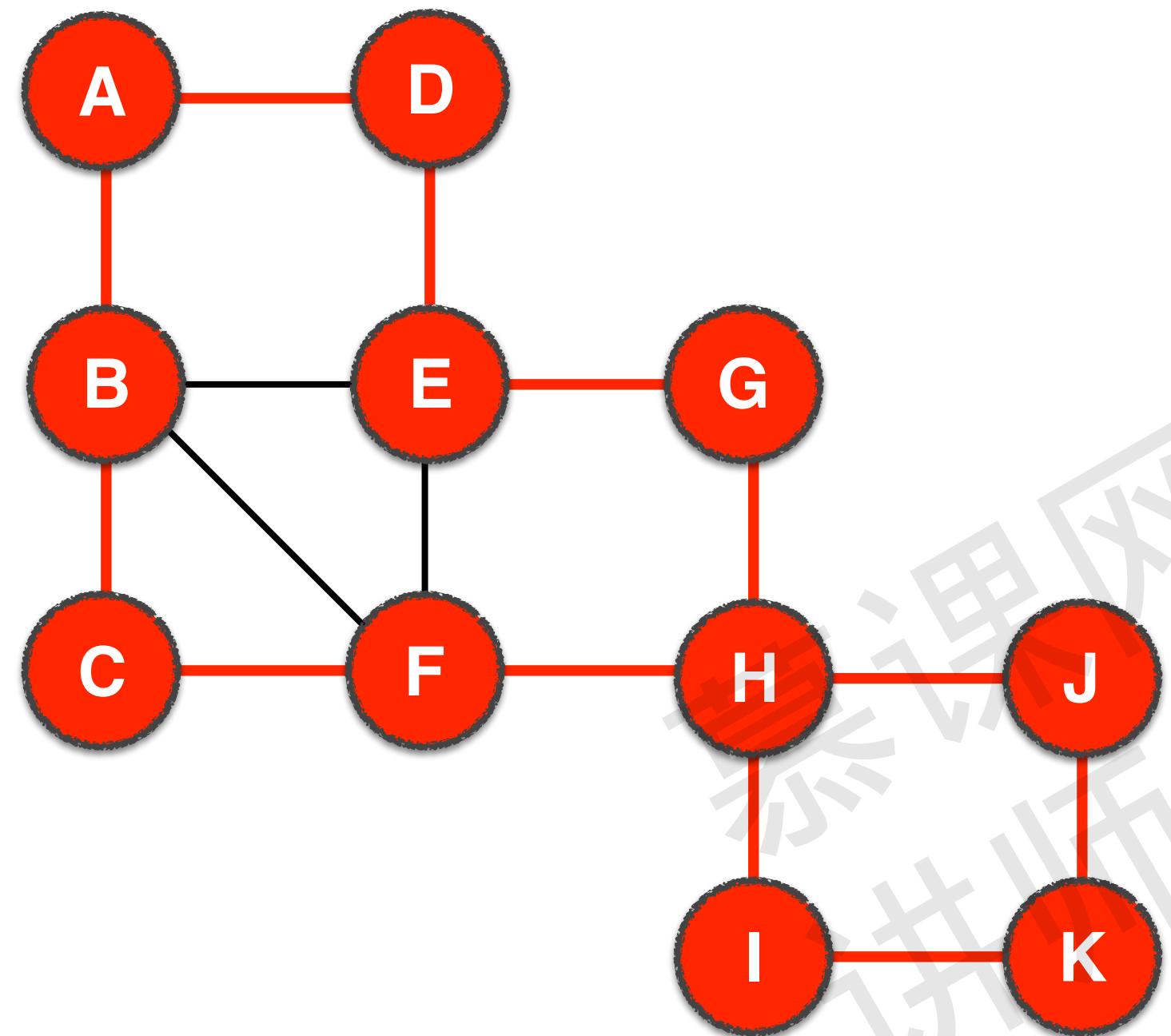
图存在欧拉回路

从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

欧拉回路存在的性质



对于无向联通图
每个点的度是偶数



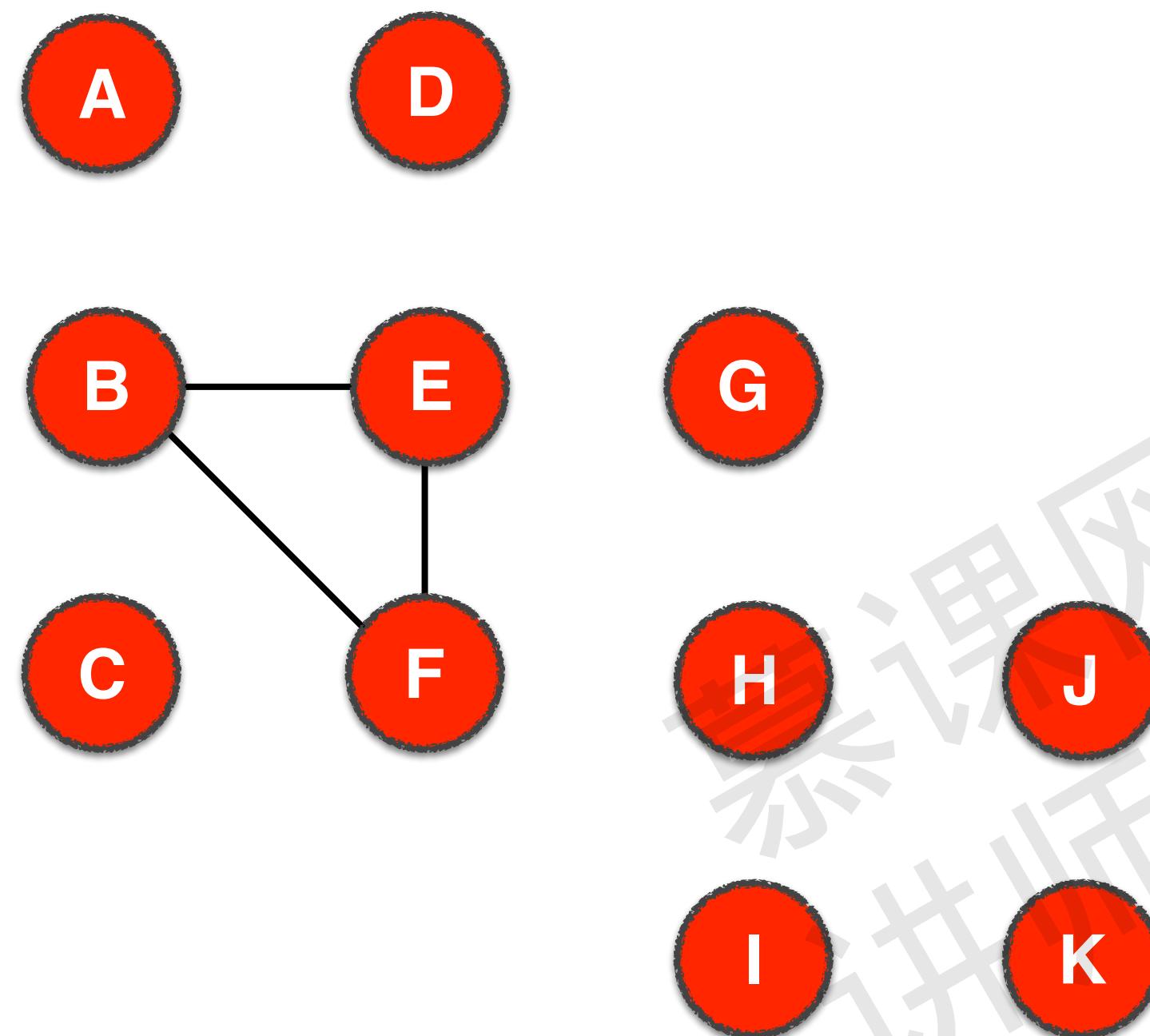
图存在欧拉回路

从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

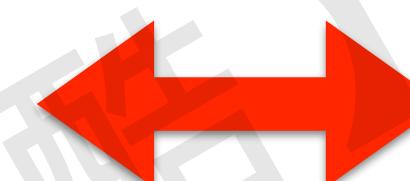
否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

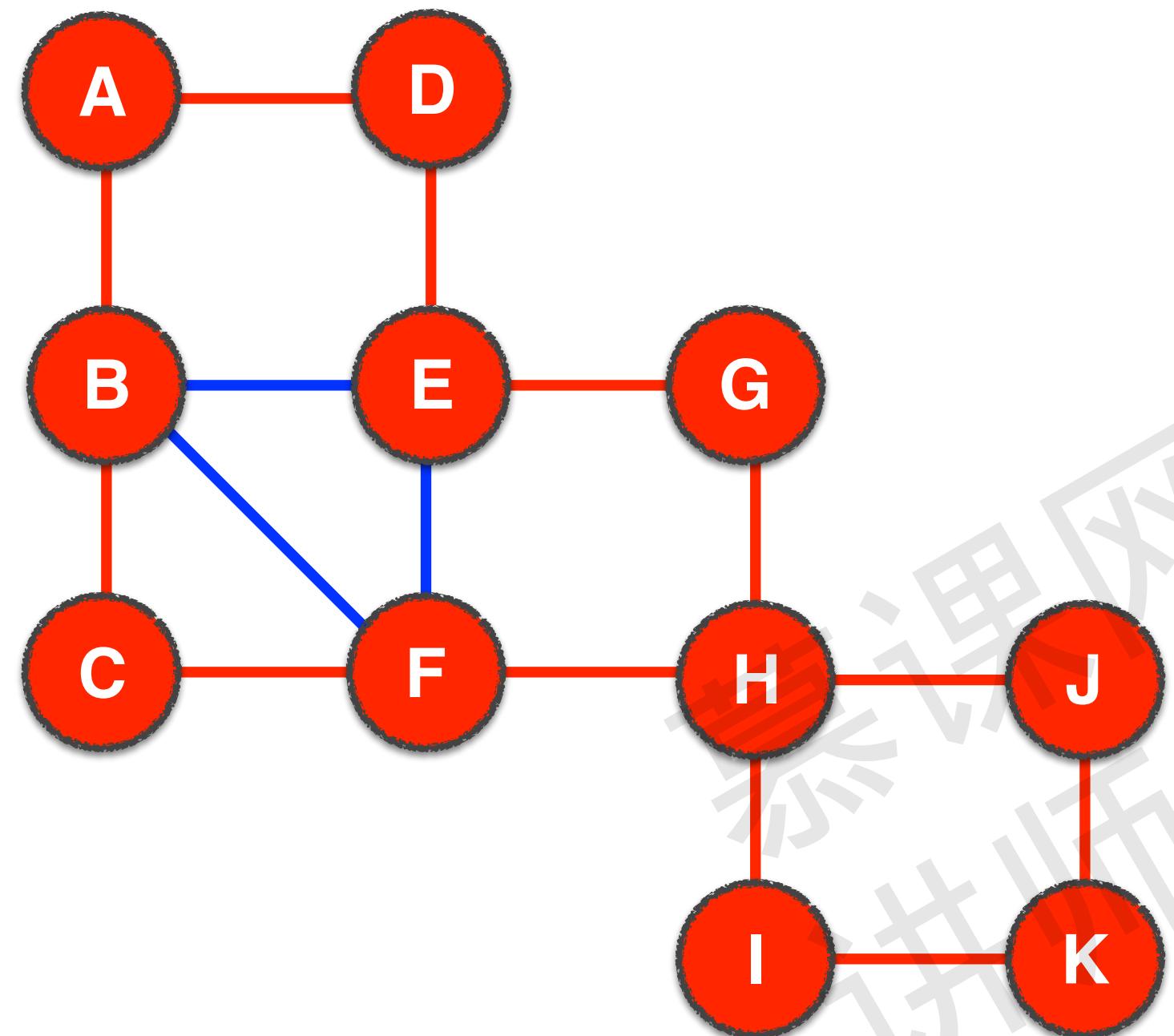
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数

图存在欧拉回路

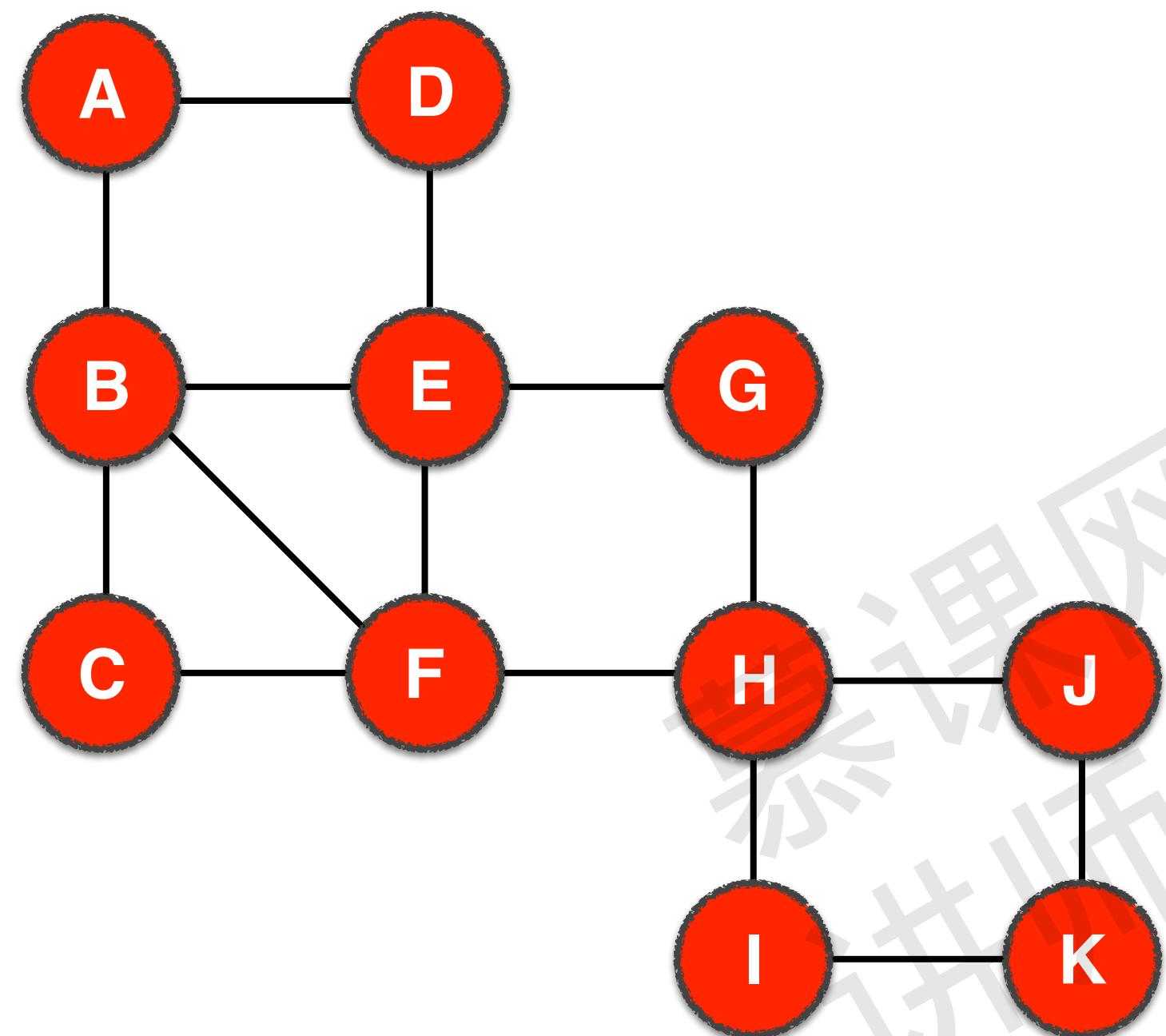
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图
每个点的度是偶数

图存在欧拉回路

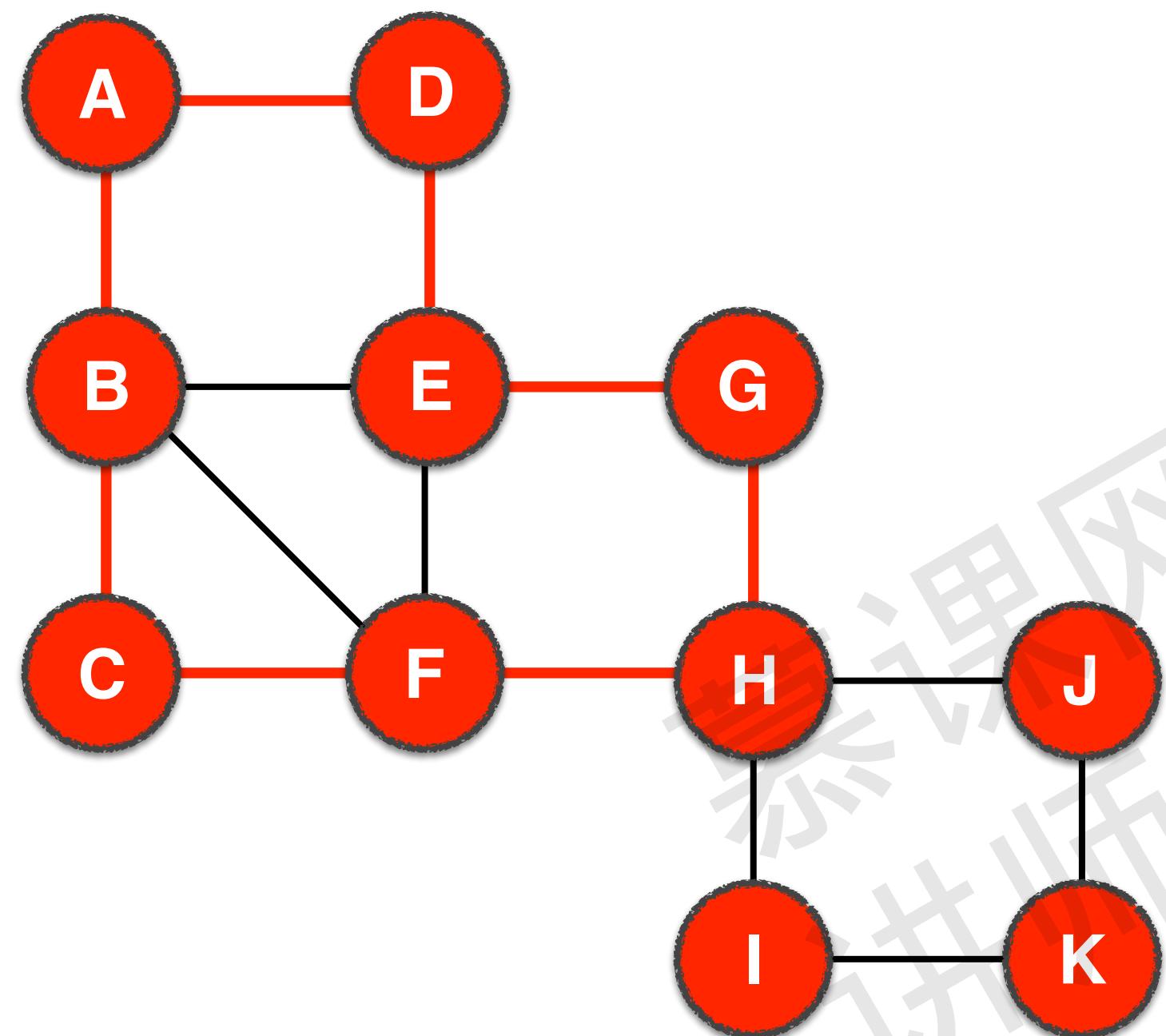
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图
每个点的度是偶数

图存在欧拉回路

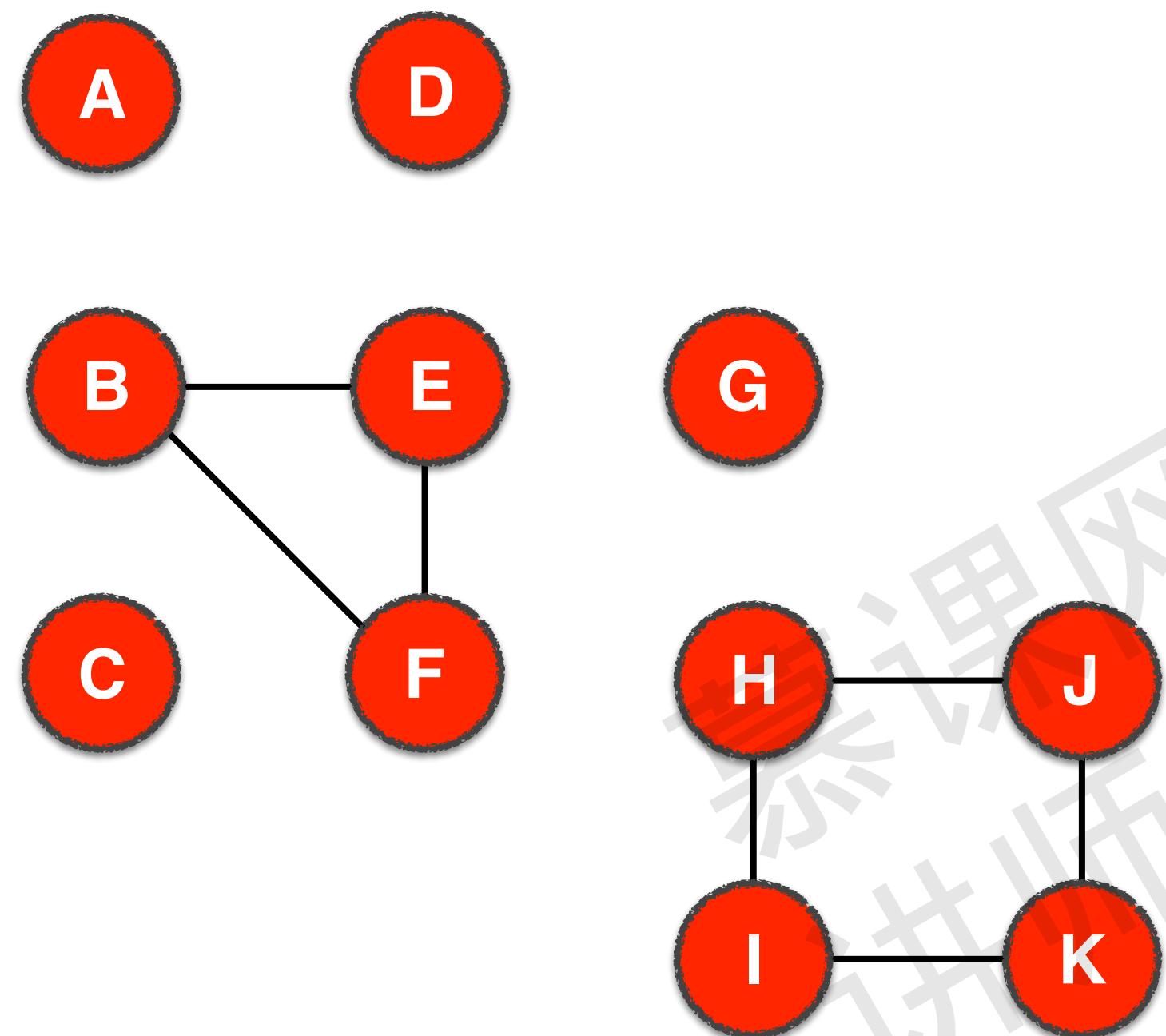
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数

图存在欧拉回路

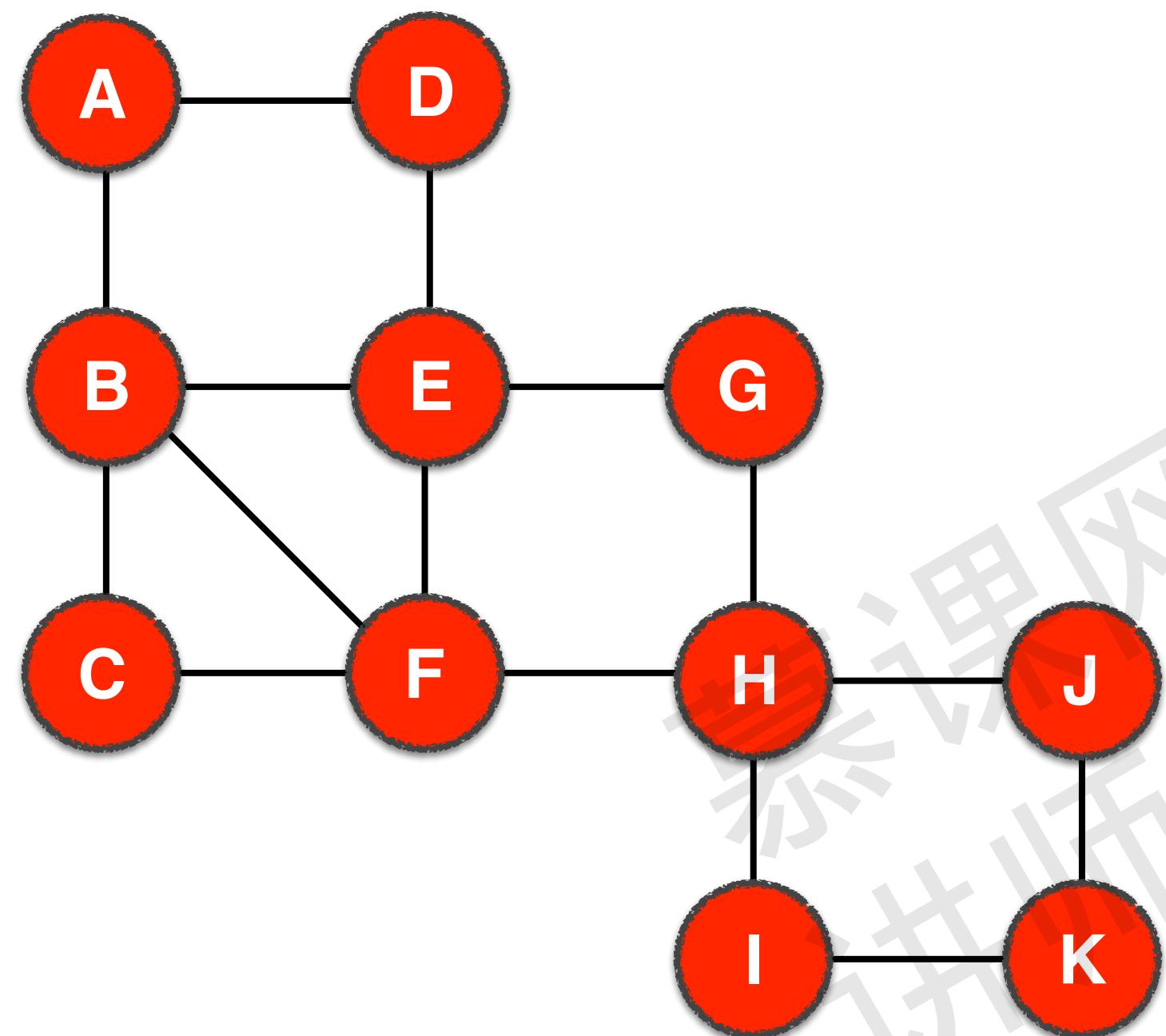
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图
每个点的度是偶数

图存在欧拉回路

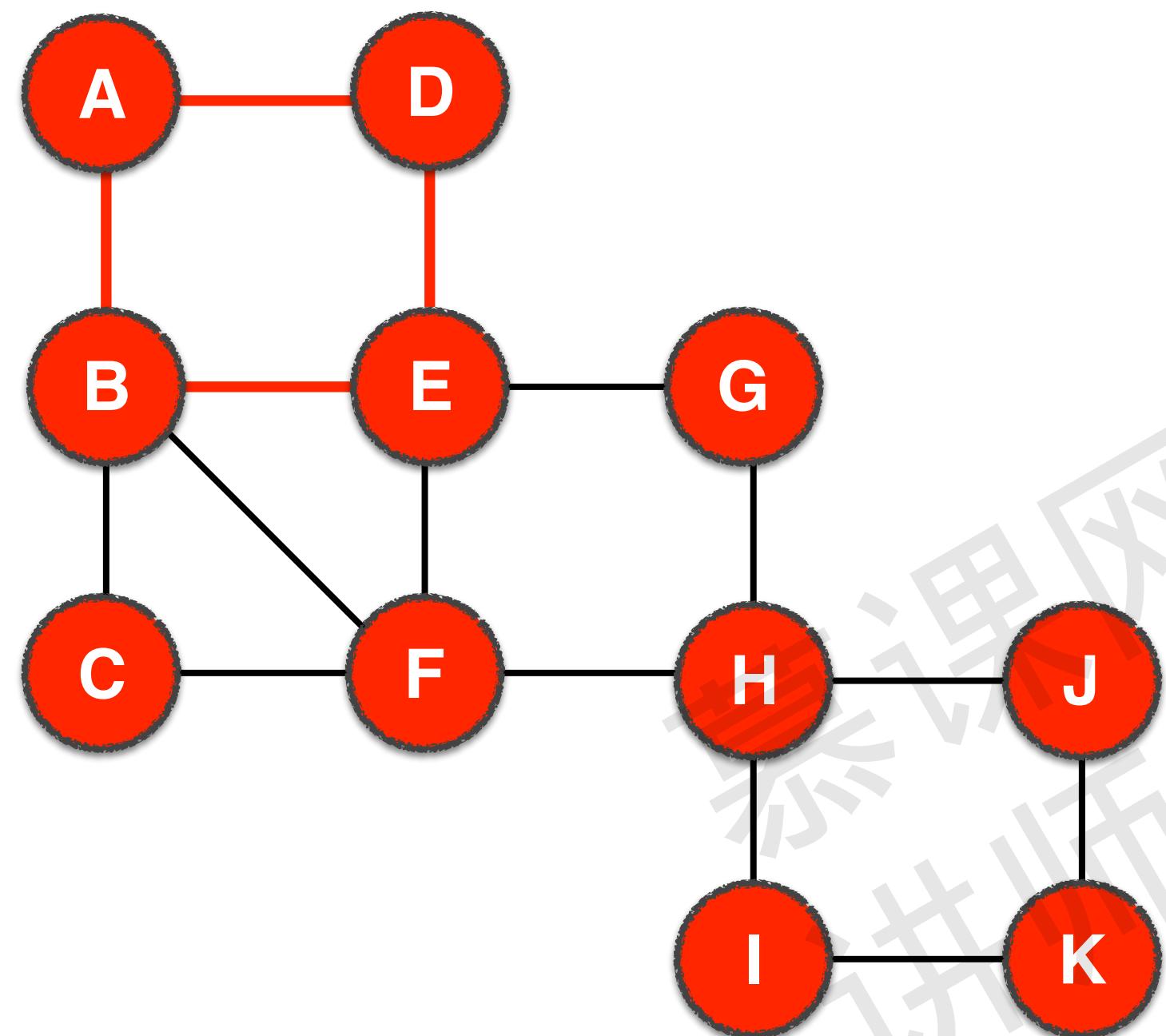
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

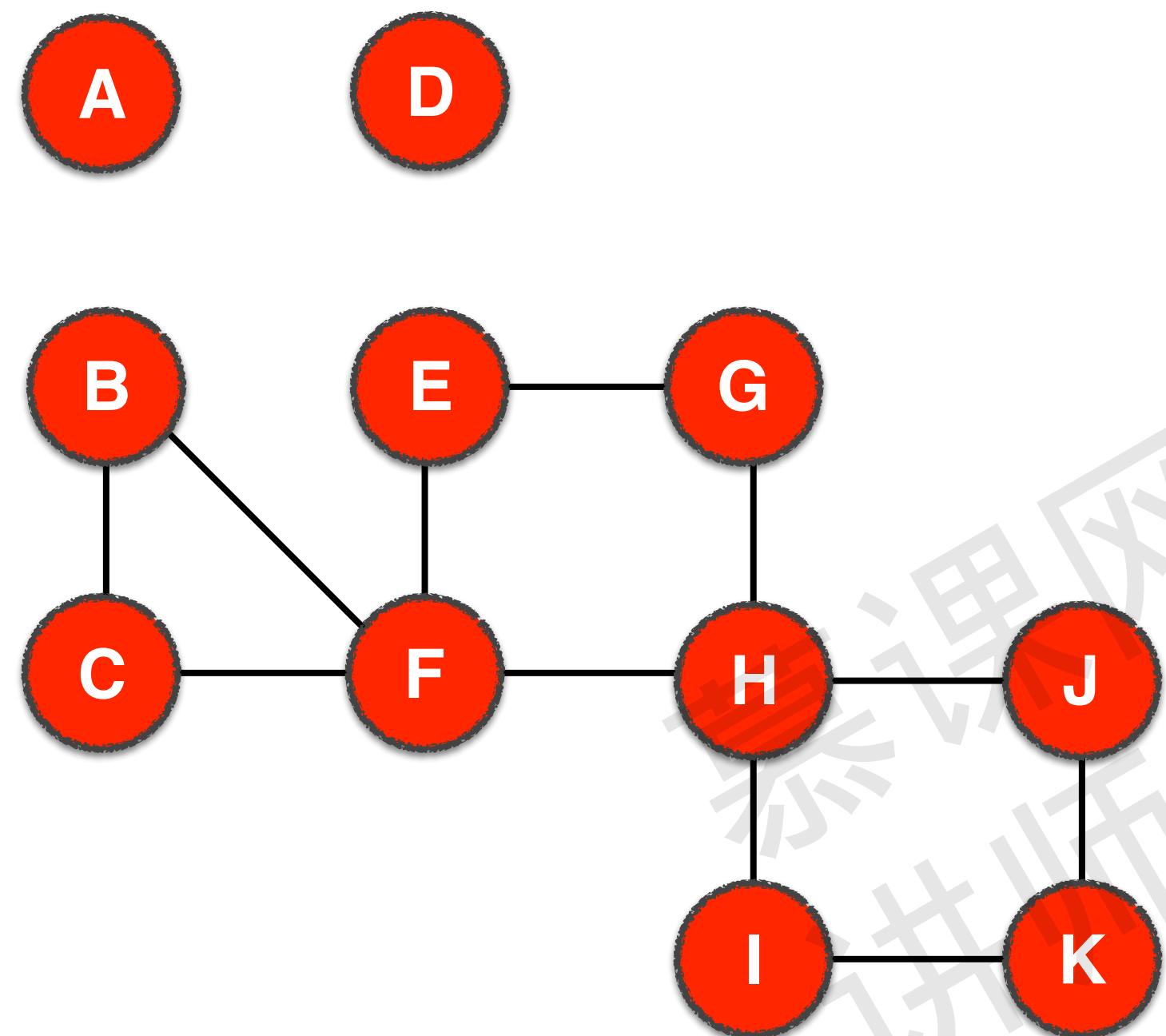
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

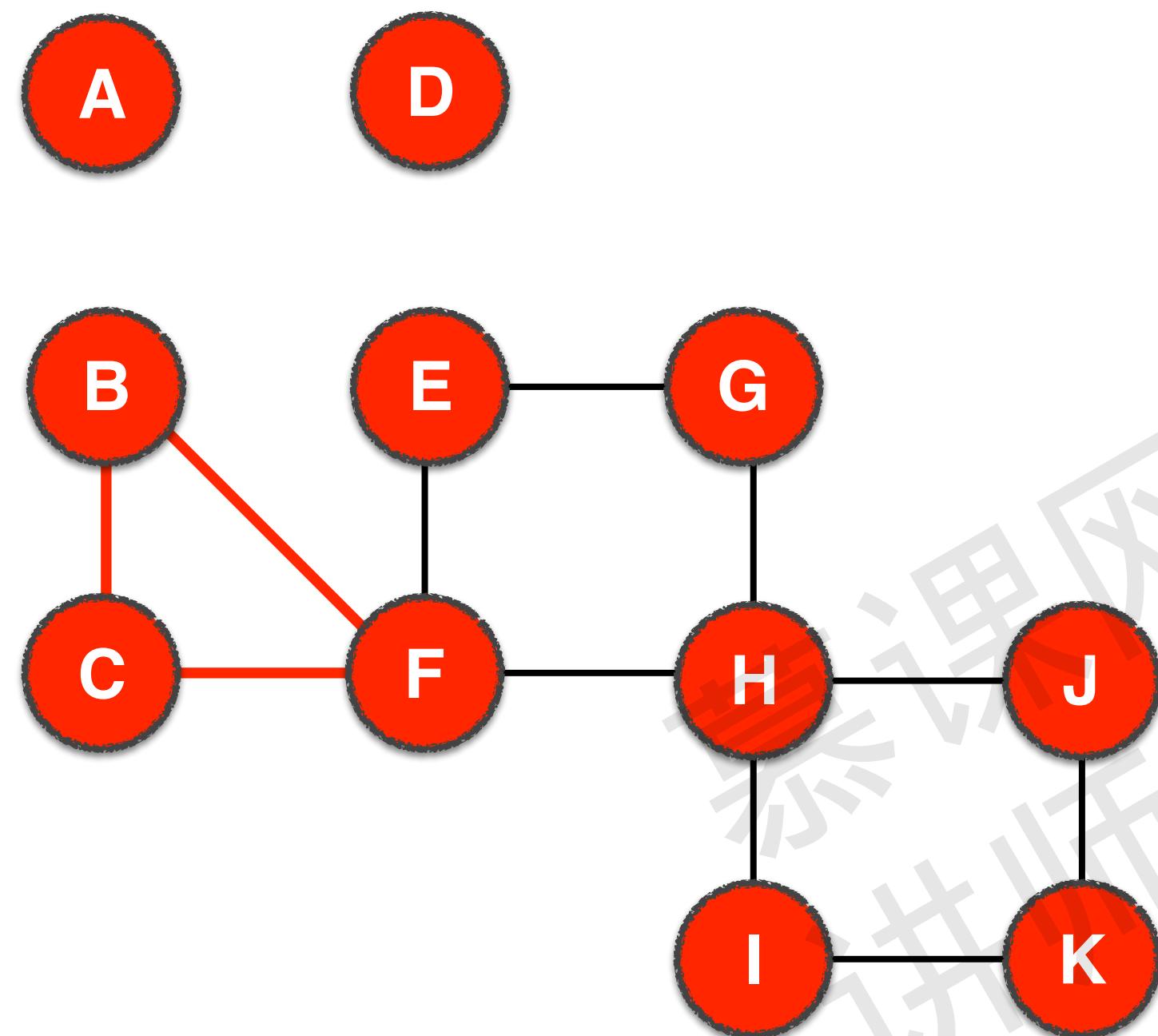
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

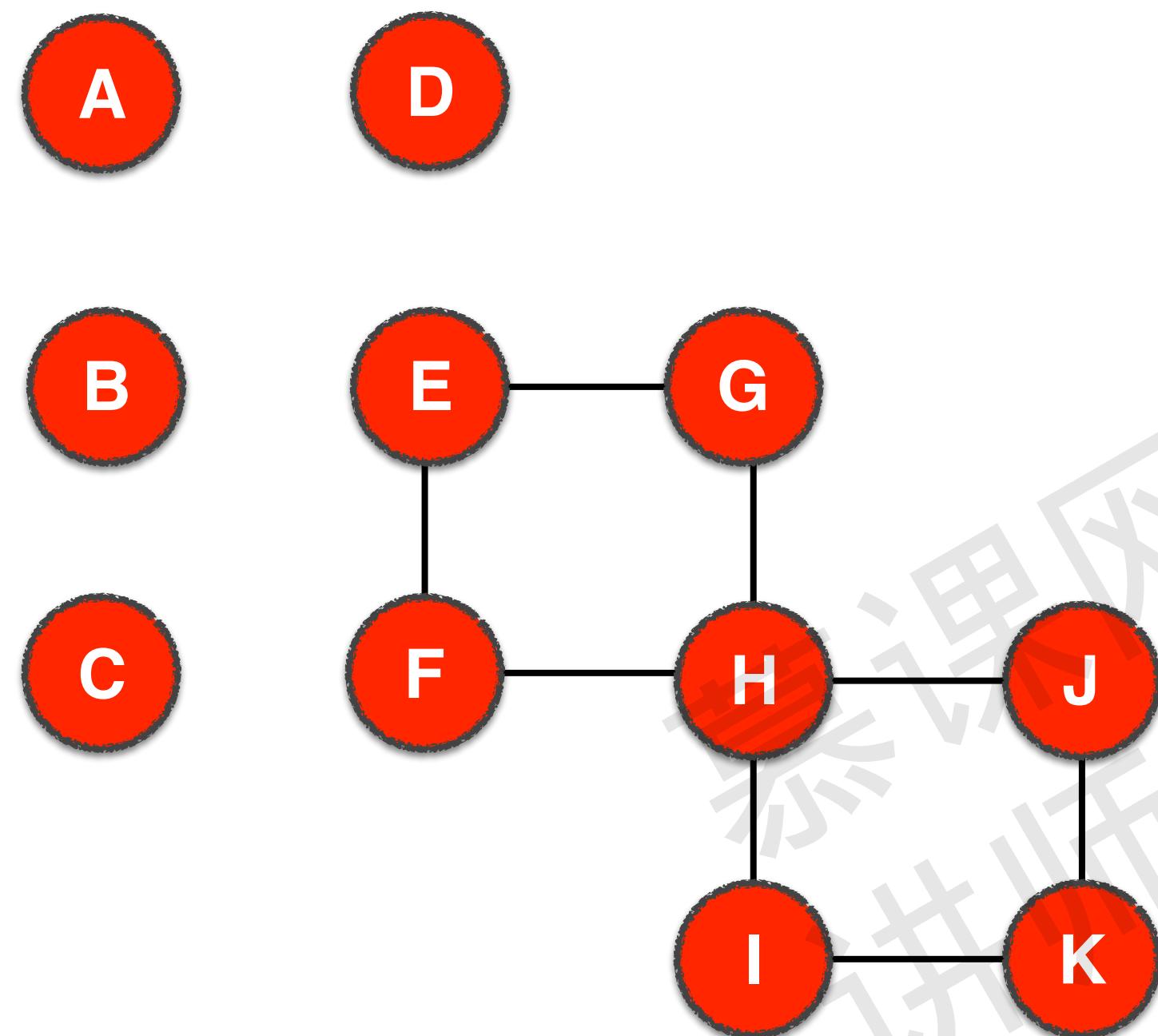
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

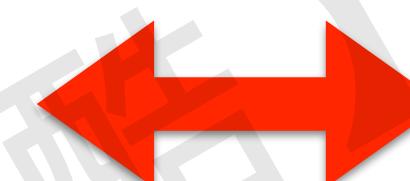
两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

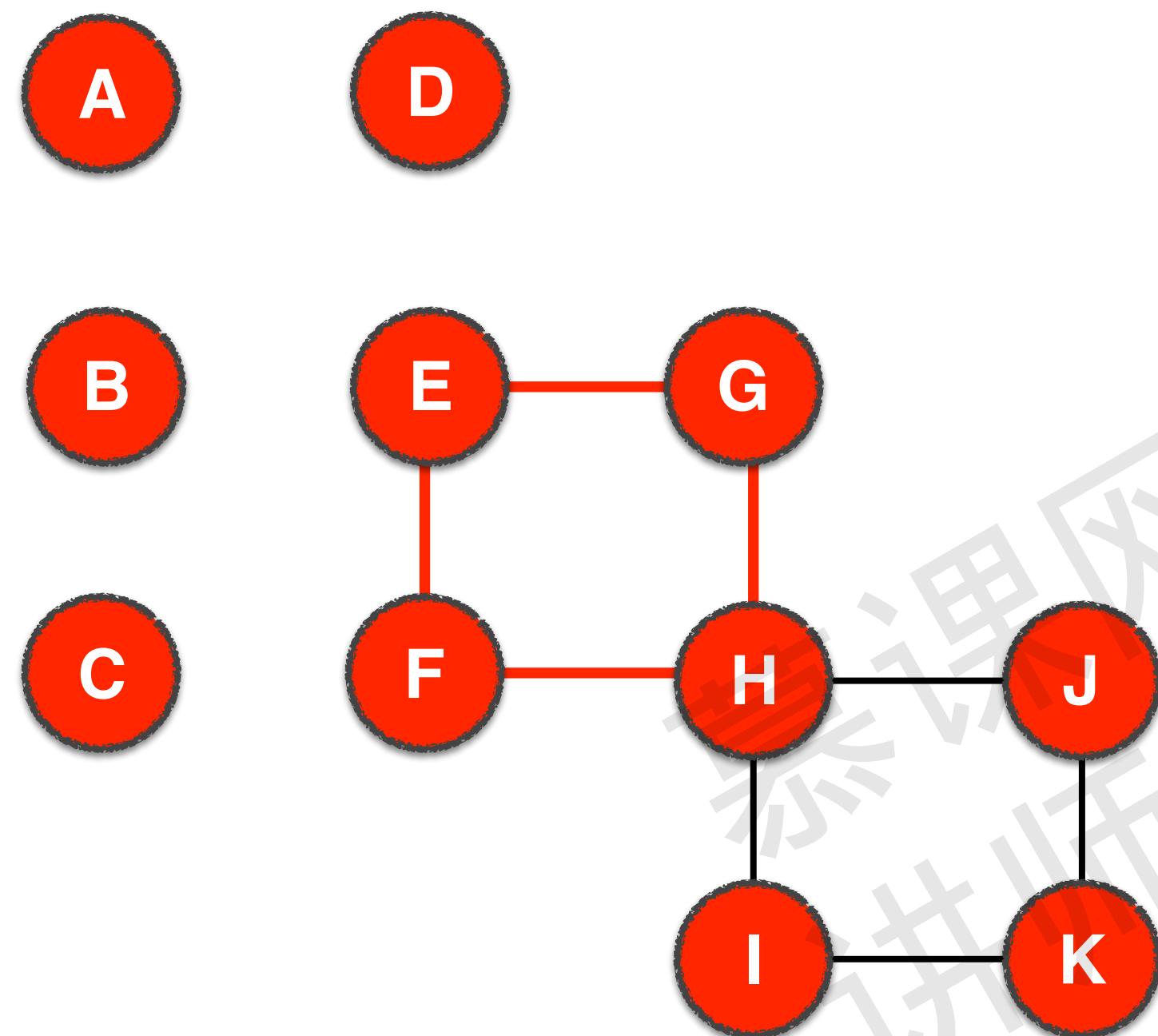
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

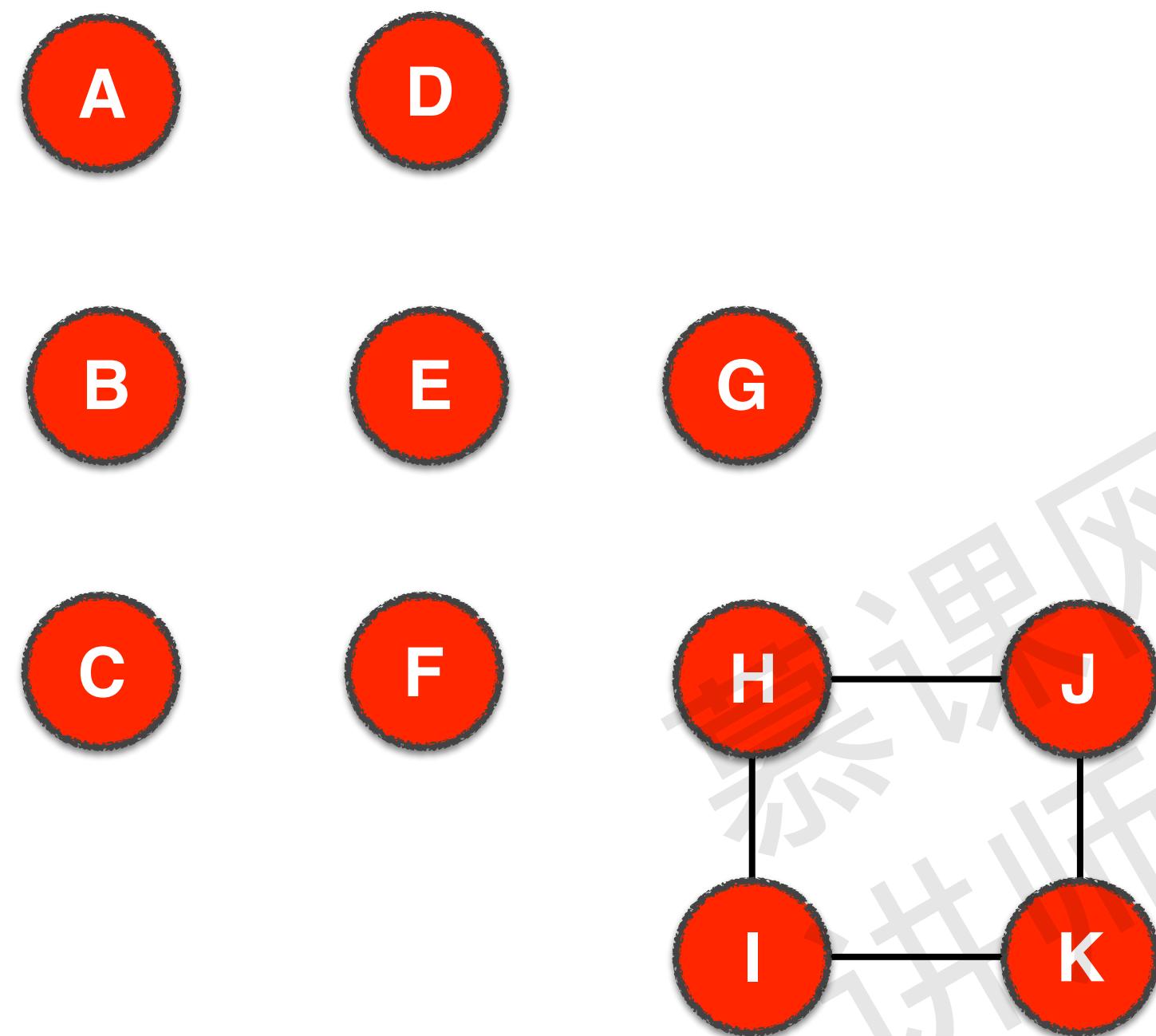
从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

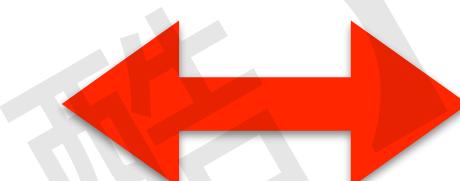
两个相连的环，一定组成一个新环

欧拉回路存在的性质



对于无向联通图

每个点的度是偶数



图存在欧拉回路

从任一点出发，先随便找一个环。

如果这个环就是原图，已经找到了欧拉回路。

否则，剩下的边一定和我们找到的环相连，
且所有顶点的度数依然是偶数，即依然存在环。

两个相连的环，一定组成一个新环

欧拉回路存在的性质

对于无向联通图

每个点的度是偶数



图存在欧拉回路

判断是否存在欧拉回路

liuyubobobo

欧拉回路存在的性质

对于无向联通图

每个点的度是偶数



图存在欧拉回路

编程实践：判断是否有欧拉回路

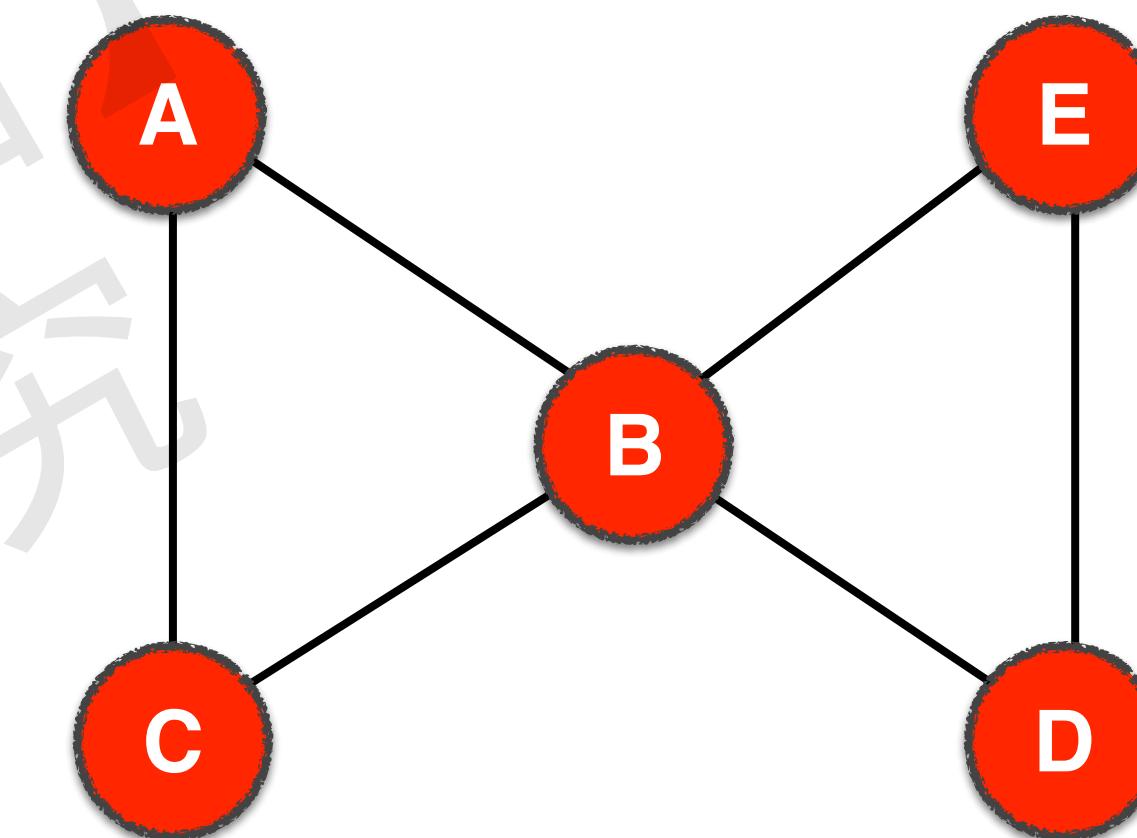
慕课网 · liuyanbo · 《玩转图论算法》

寻找欧拉回路的三个算法

liuyubobobo

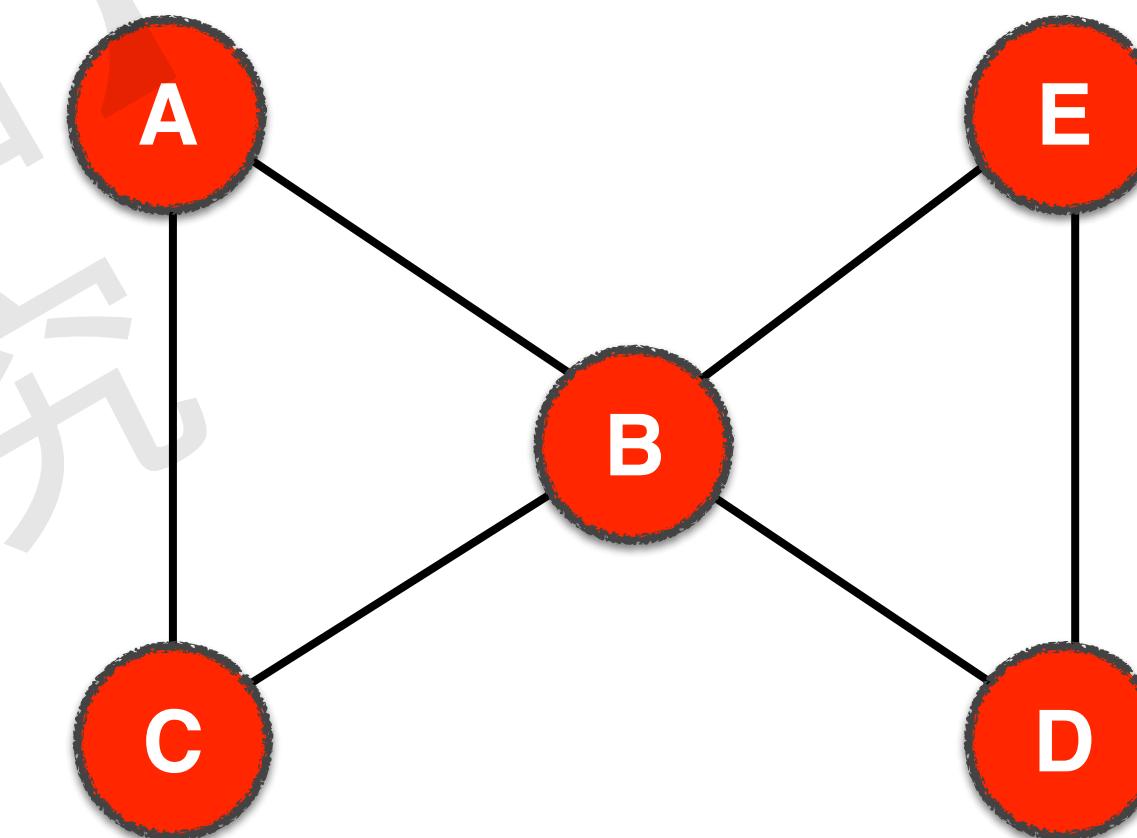
寻找欧拉回路

回溯法



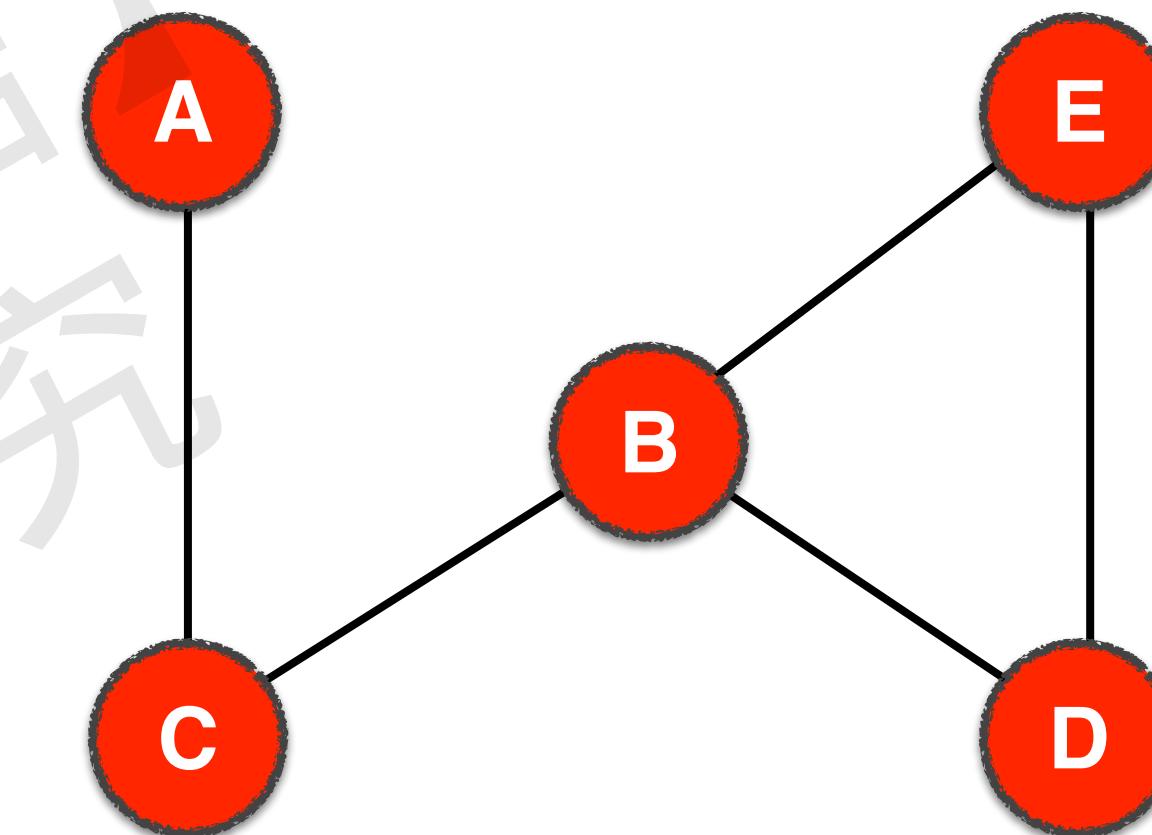
寻找欧拉回路

回溯法



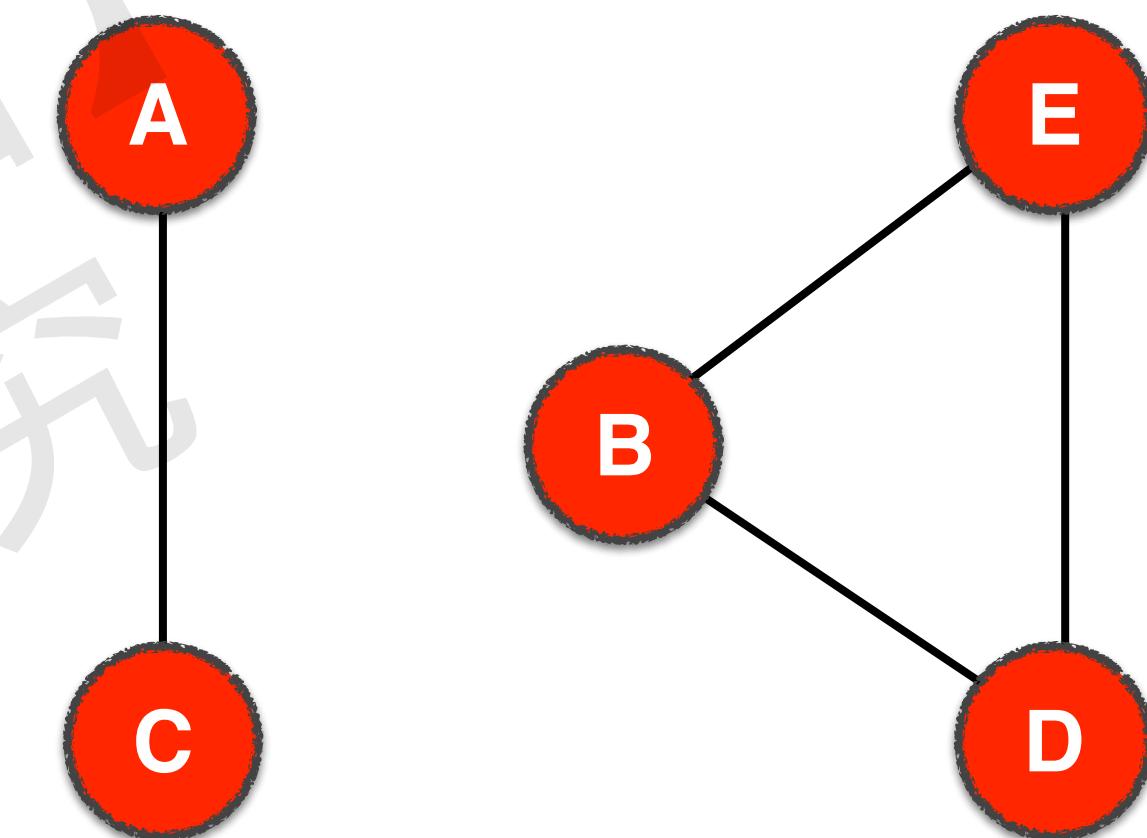
寻找欧拉回路

回溯法



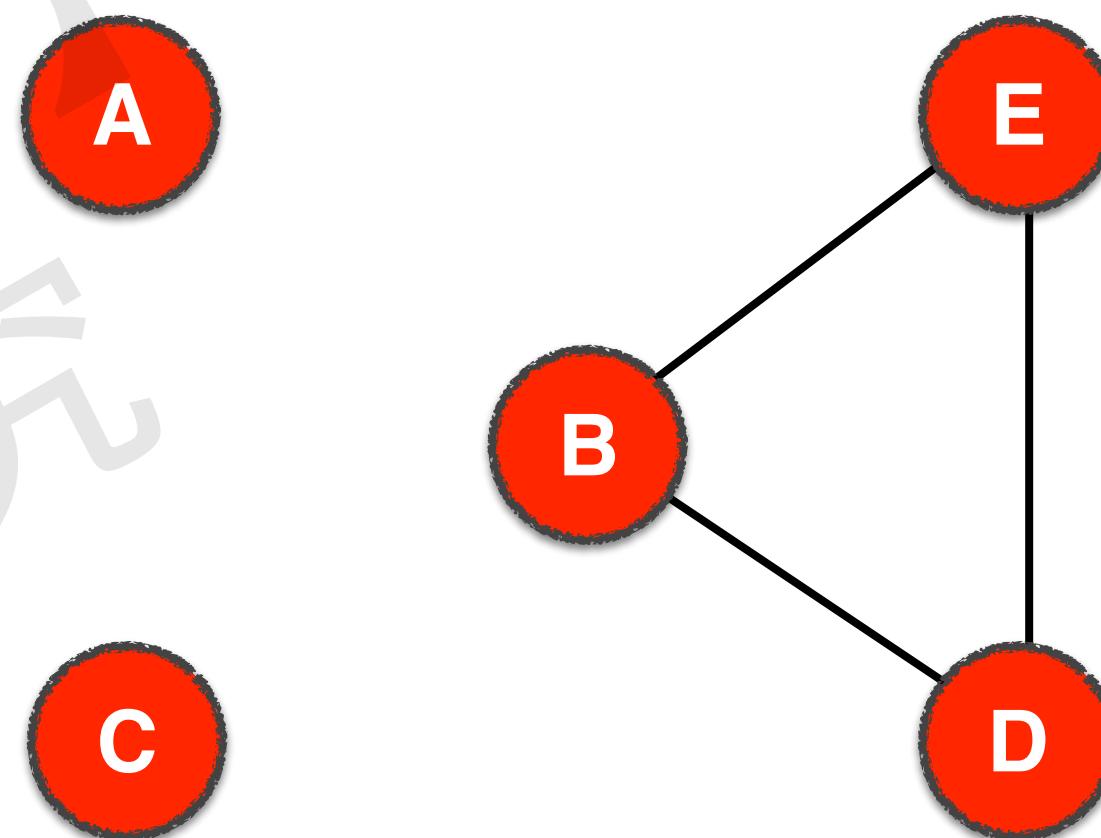
寻找欧拉回路

回溯法



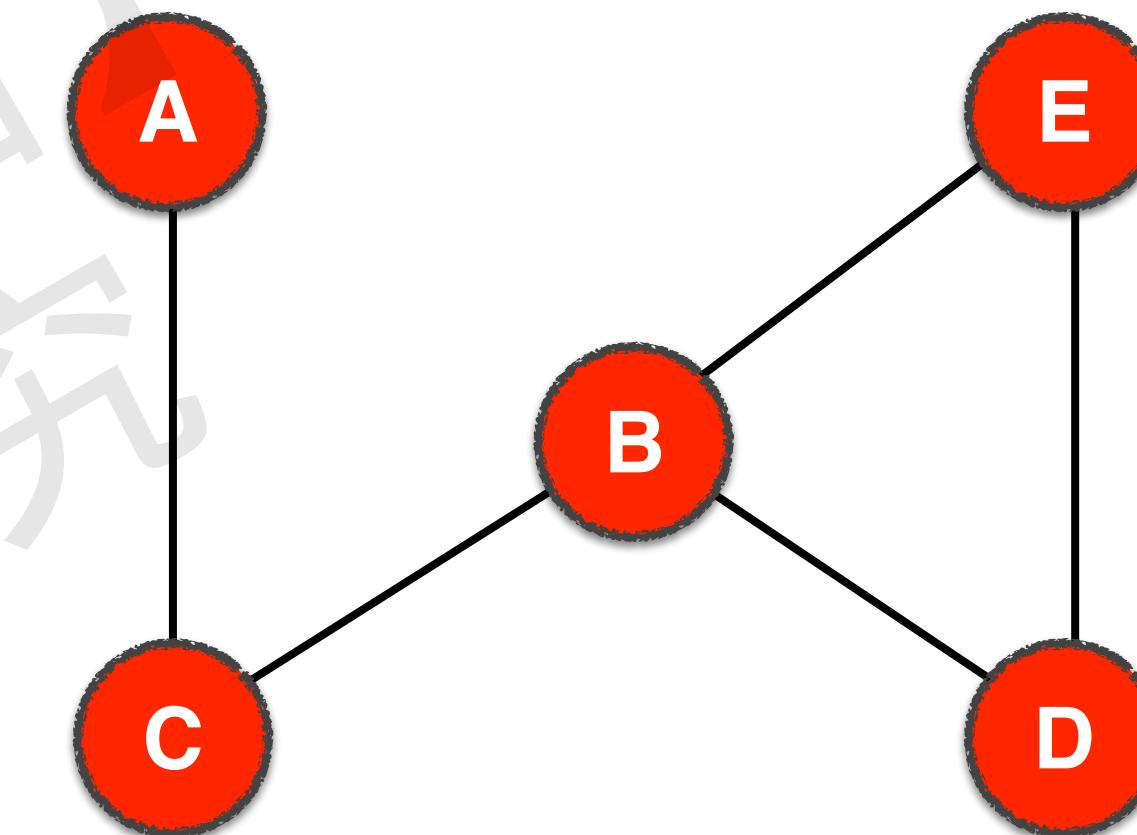
寻找欧拉回路

回溯法



寻找欧拉回路

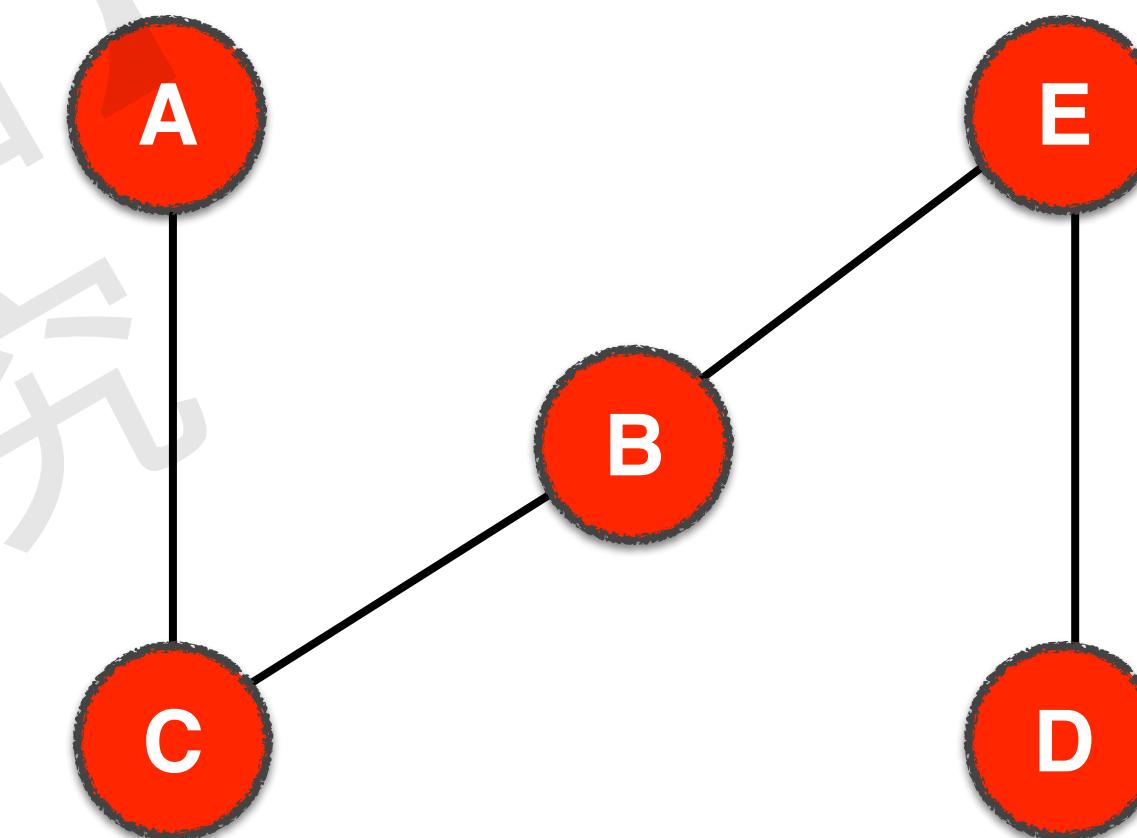
回溯法



寻找欧拉回路

回溯法

指数级

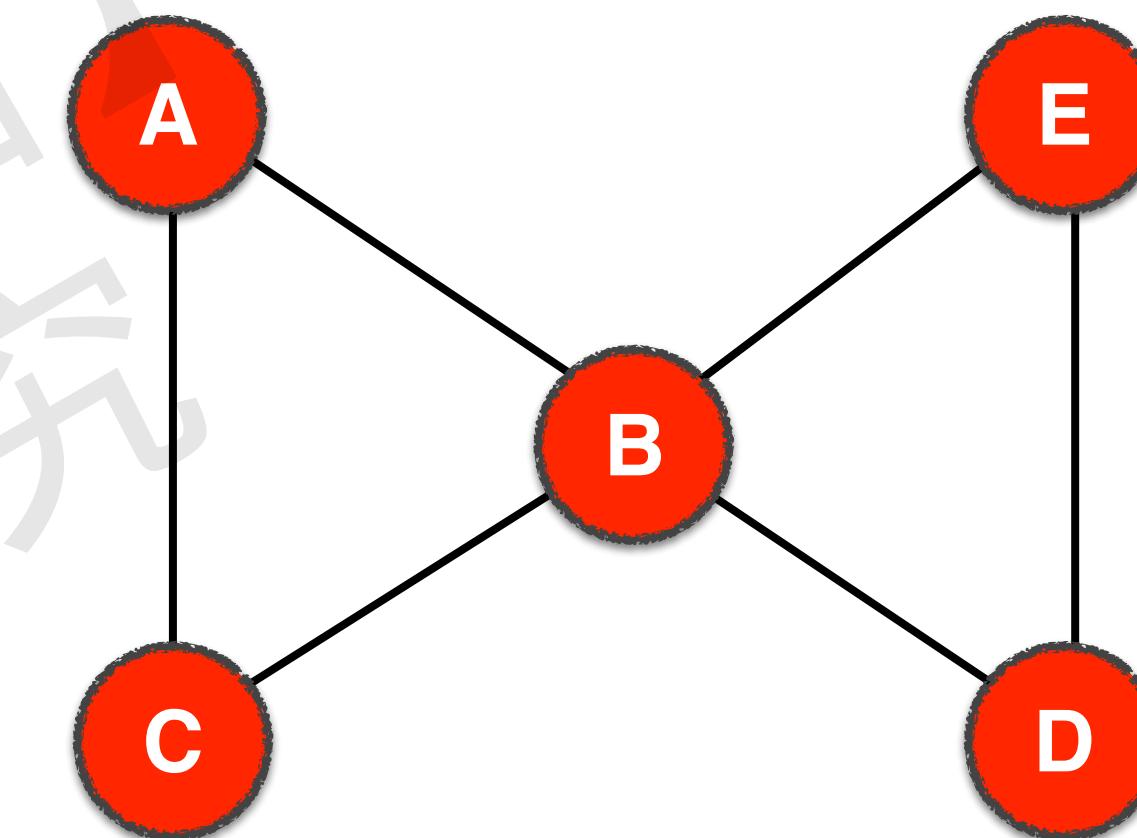


寻找欧拉回路

回溯法

指数级

Fleury 算法

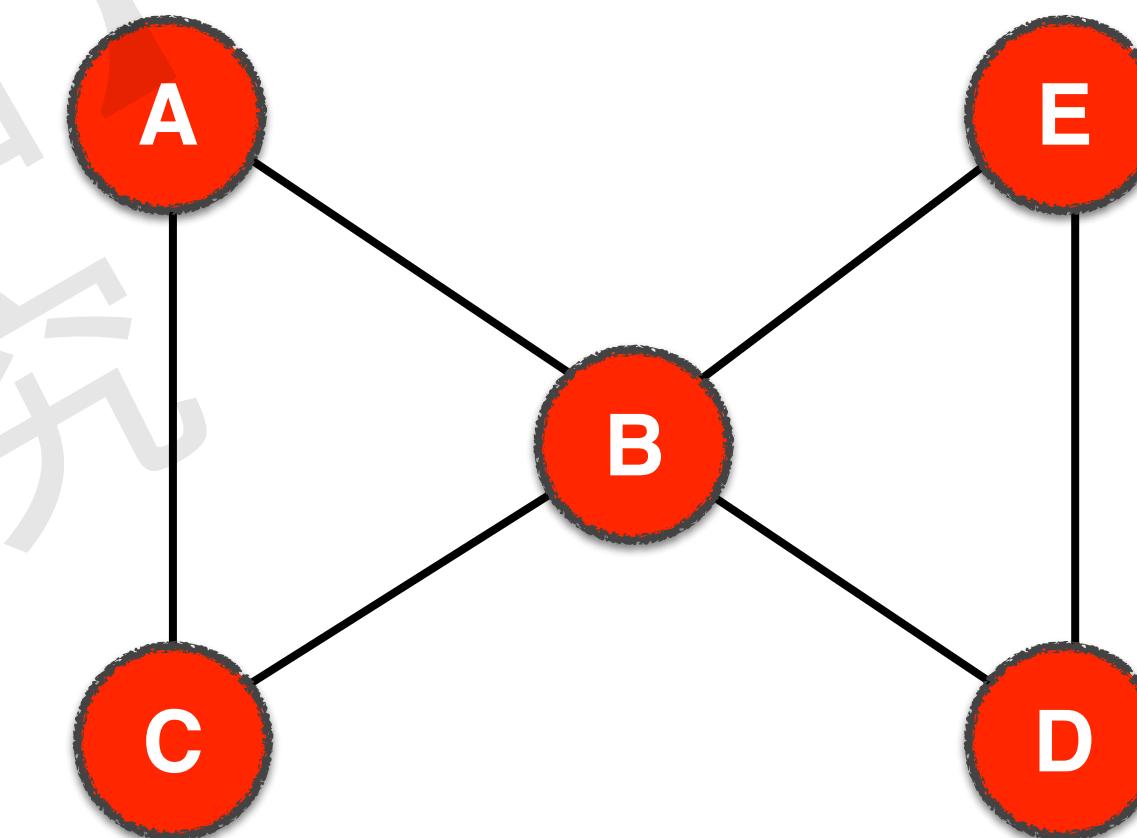


寻找欧拉回路

回溯法

指数级

Fleury 算法

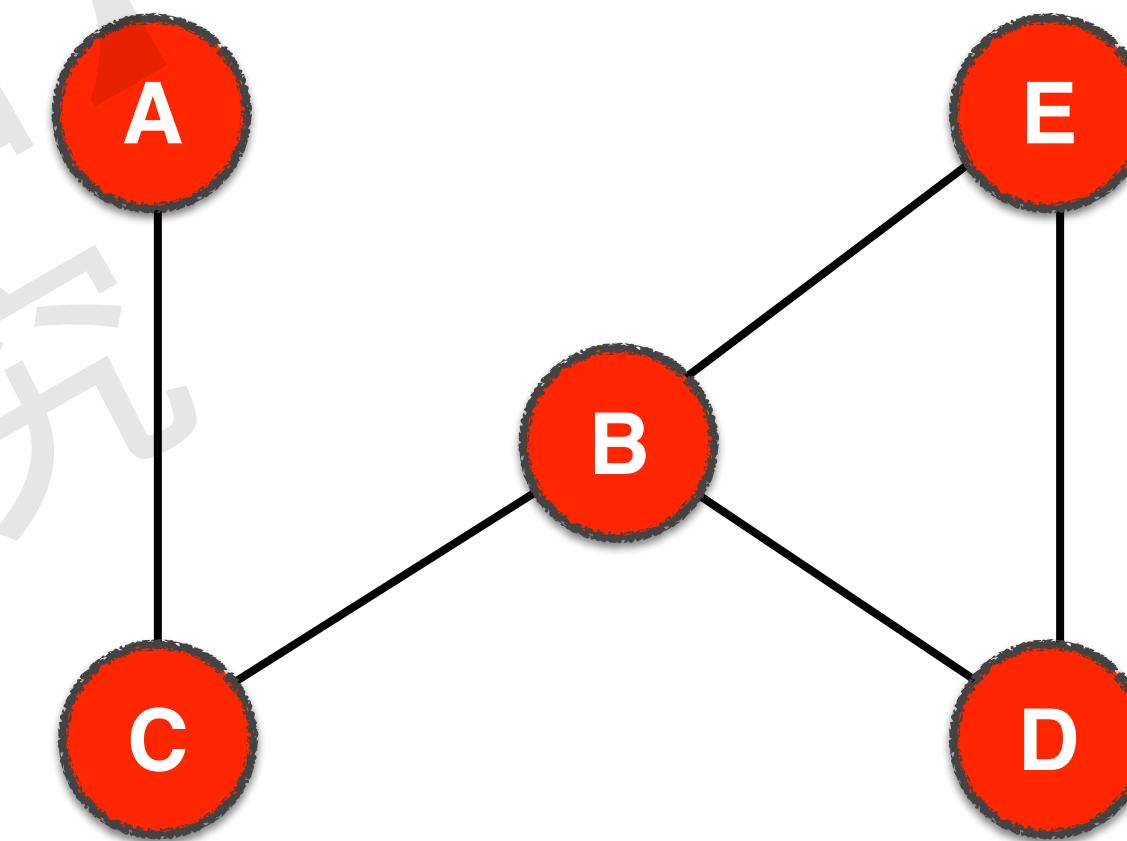


寻找欧拉回路

回溯法

指数级

Fleury 算法



寻找欧拉回路

回溯法

指数级

Fleury 算法

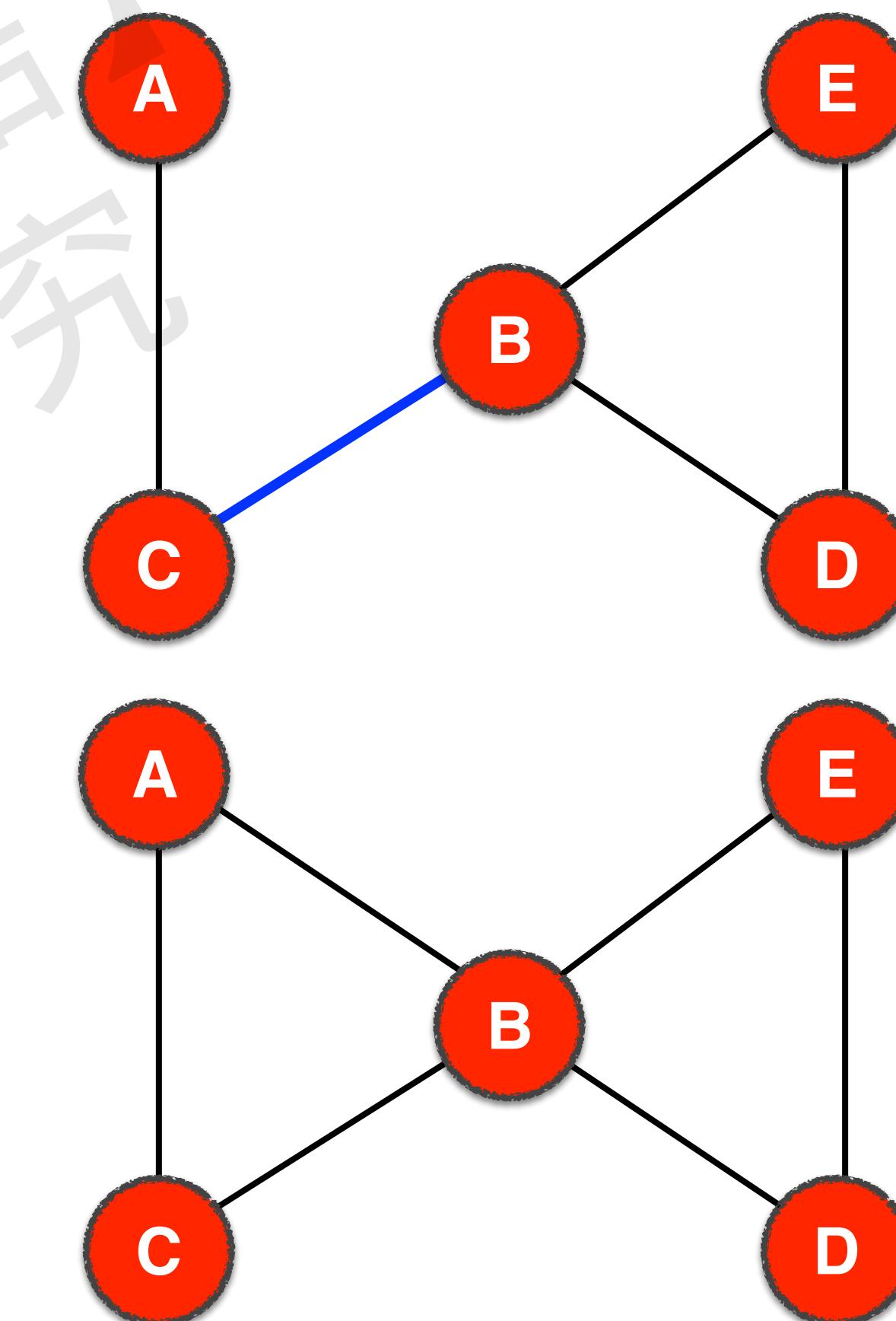
有多条边的时候，不走桥

对每一个临边，判断一下桥

不能预处理

$O((V+E)^2)$

$O(E^2)$



寻找欧拉回路

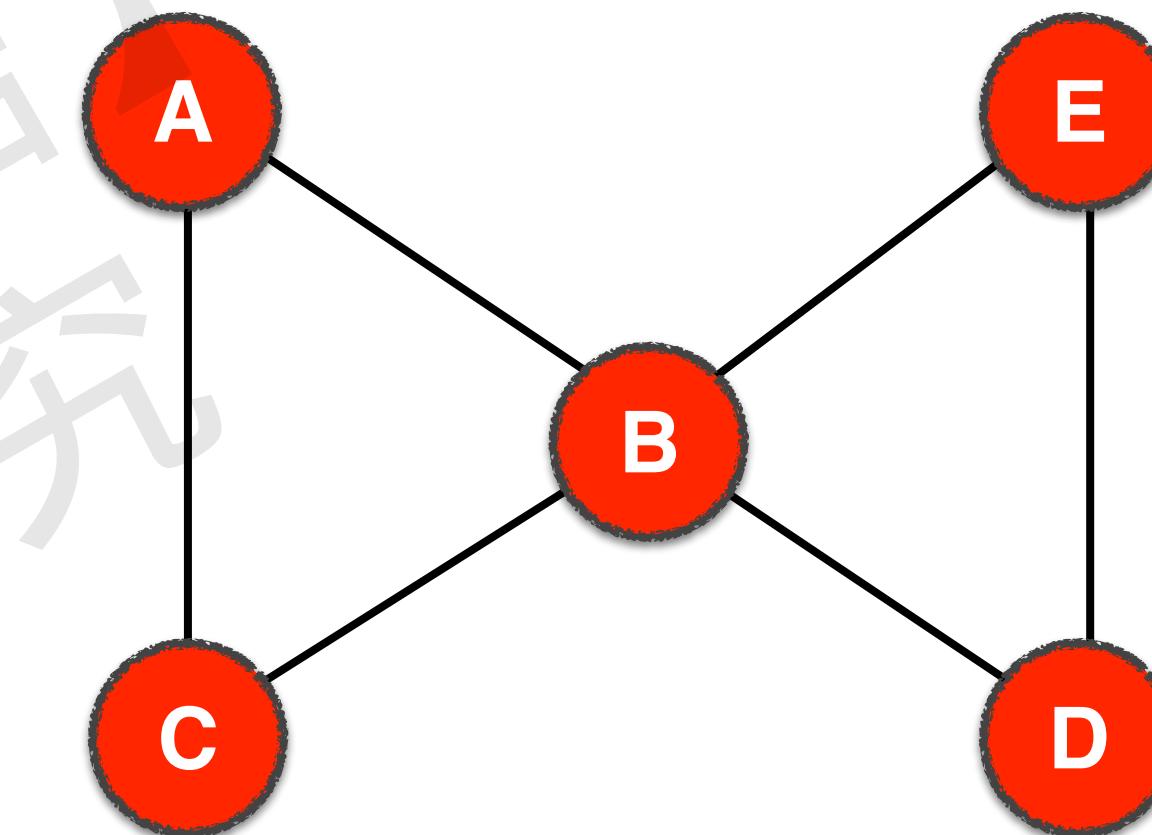
回溯法

指数级

Fleury 算法

$O(E^2)$

Hierholzer 算法



寻找欧拉回路

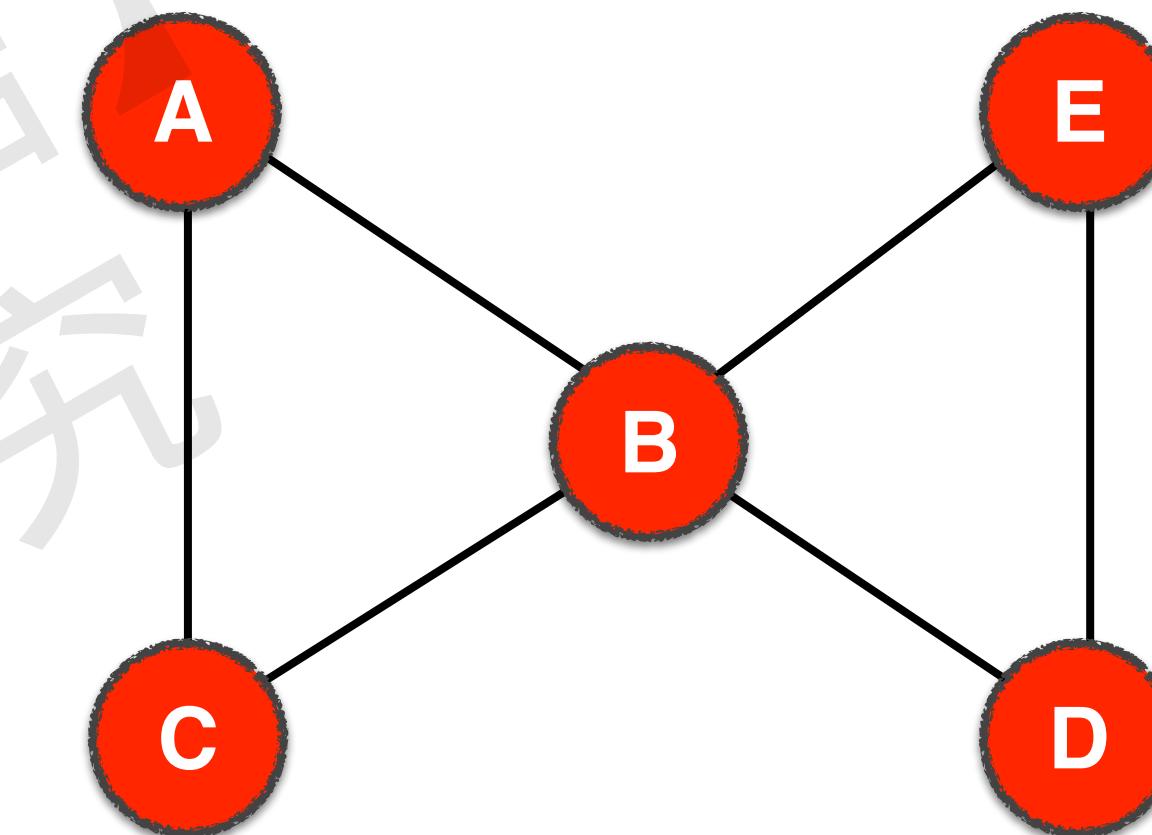
回溯法

指数级

Fleury 算法

$O(E^2)$

Hierholzer 算法



寻找欧拉回路

回溯法

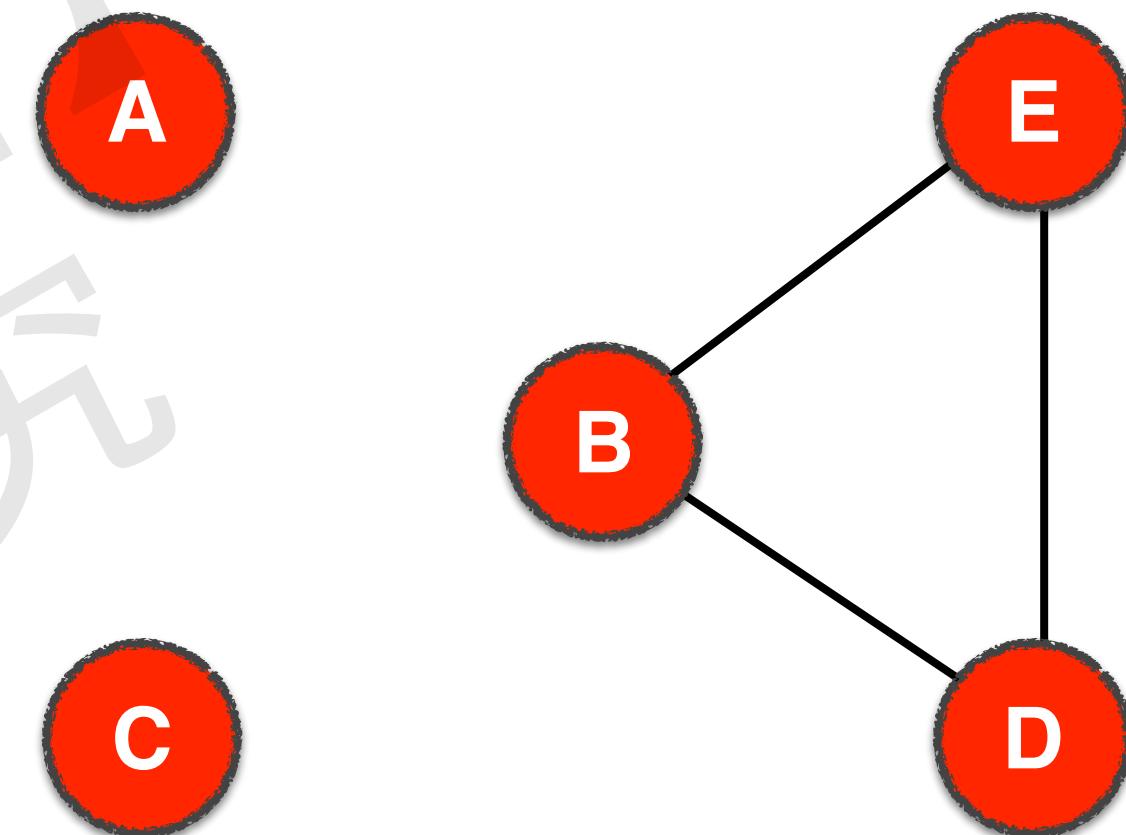
指数级

Fleury 算法

$O(E^2)$

Hierholzer 算法

$O(V + E)$



Hierholzer 算法详解

liuyubobobo

Hierholzer 算法详解

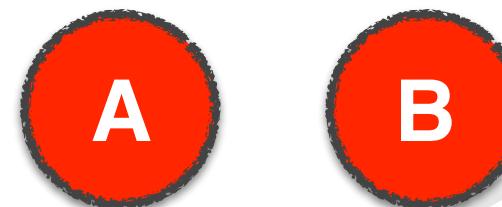
也可以使用递归完成

这个课程选择使用非递归完成

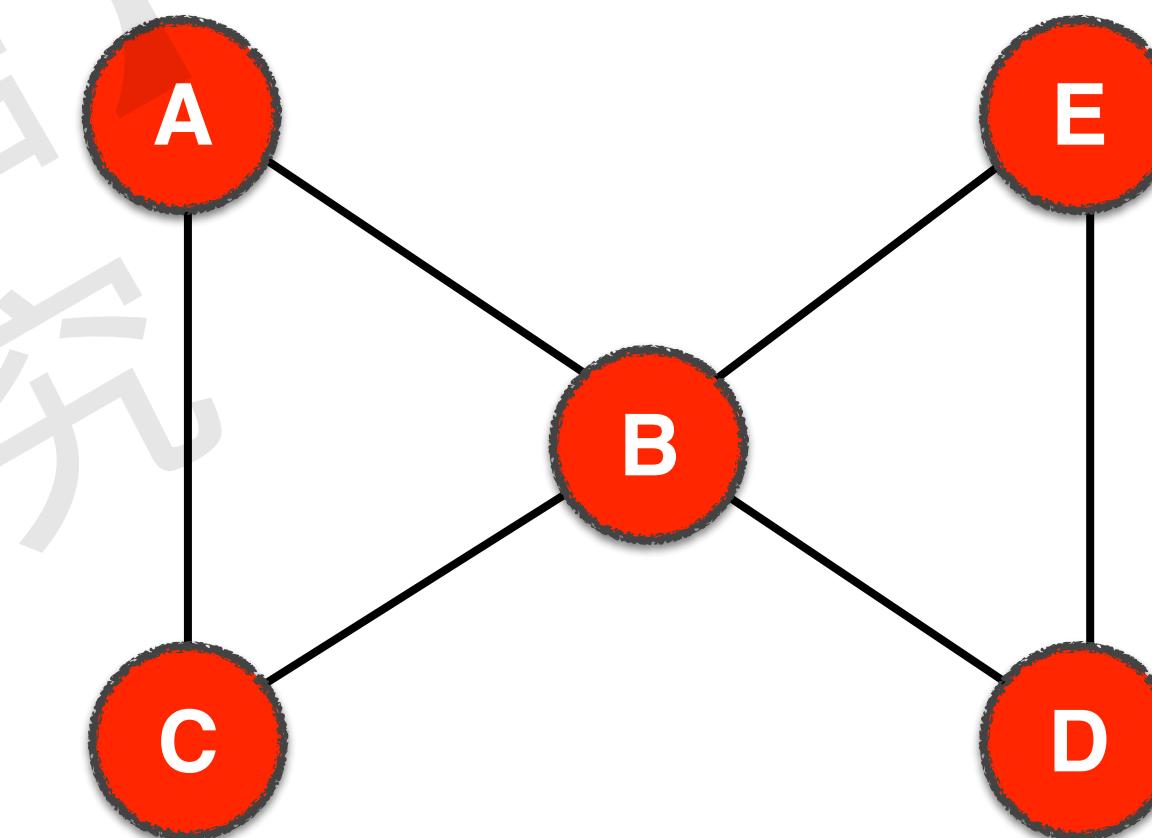
Hierholzer 算法详解

使用两个栈

curPath:



loop:



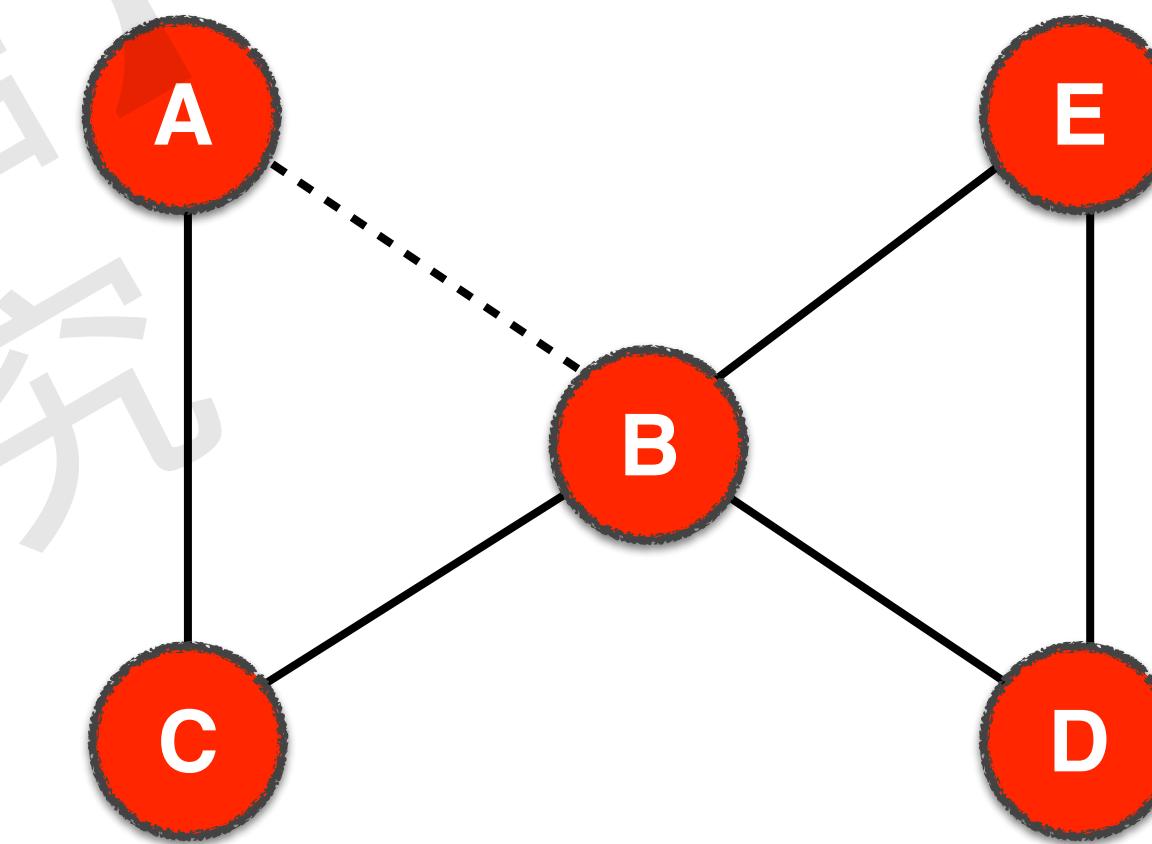
Hierholzer 算法详解

使用两个栈

curPath:



loop:



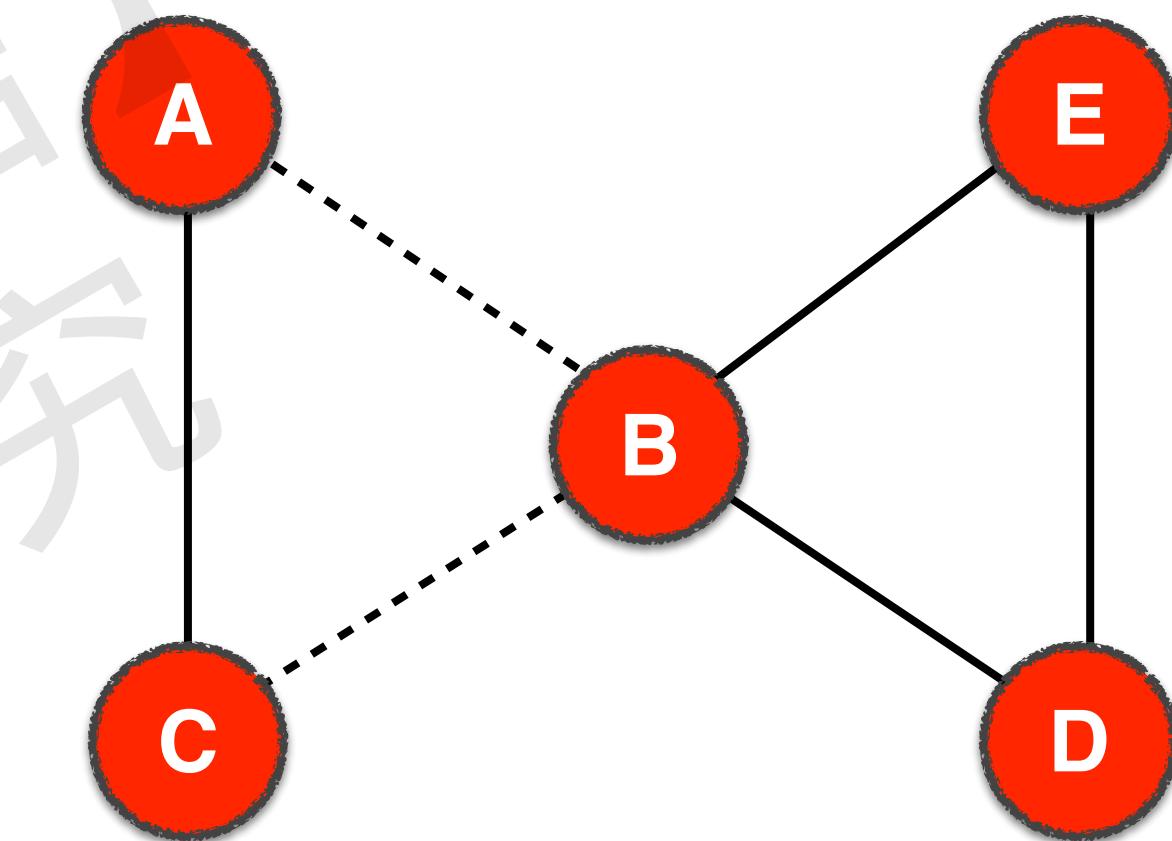
Hierholzer 算法详解

使用两个栈

curPath:



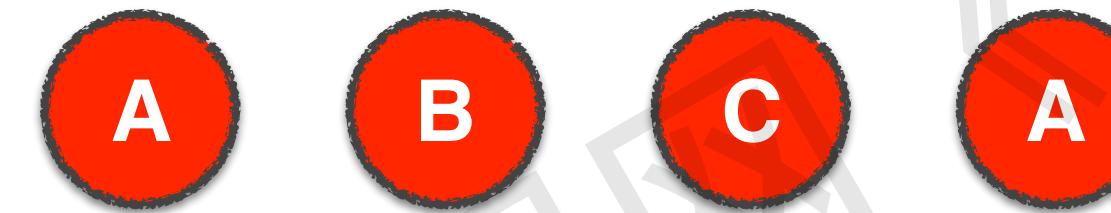
loop:



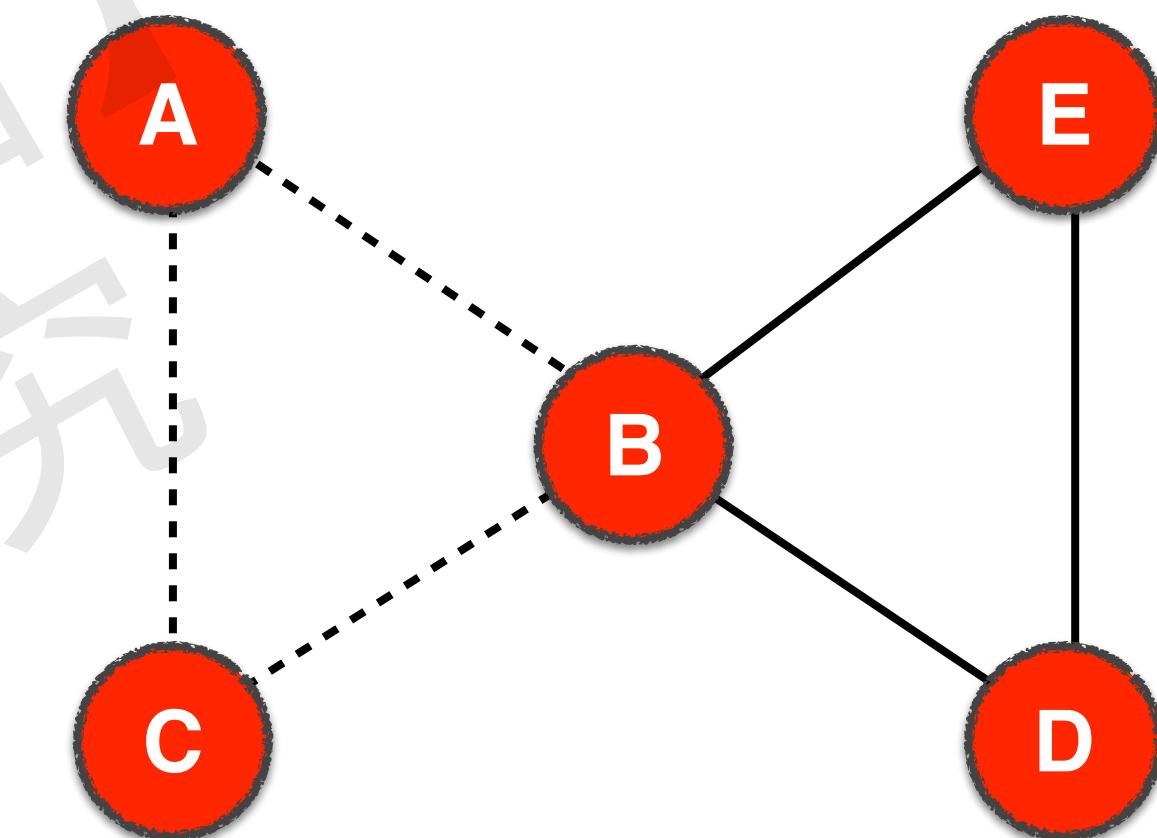
Hierholzer 算法详解

使用两个栈

curPath:



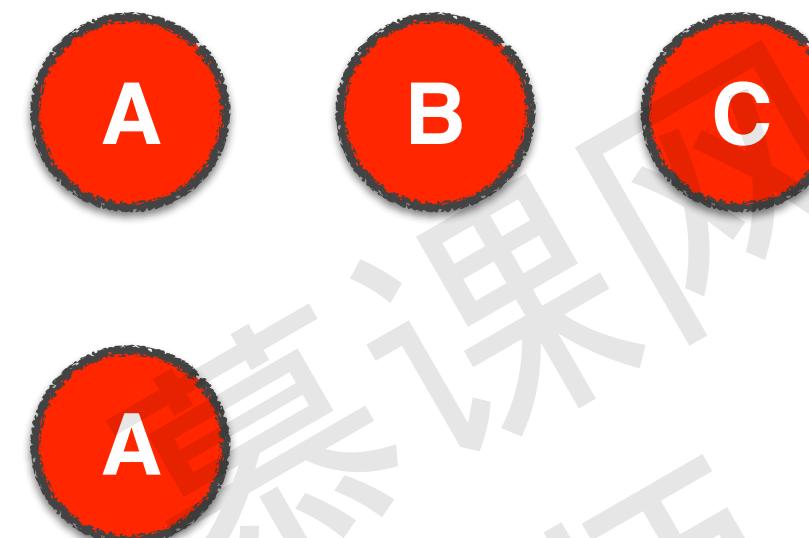
loop:



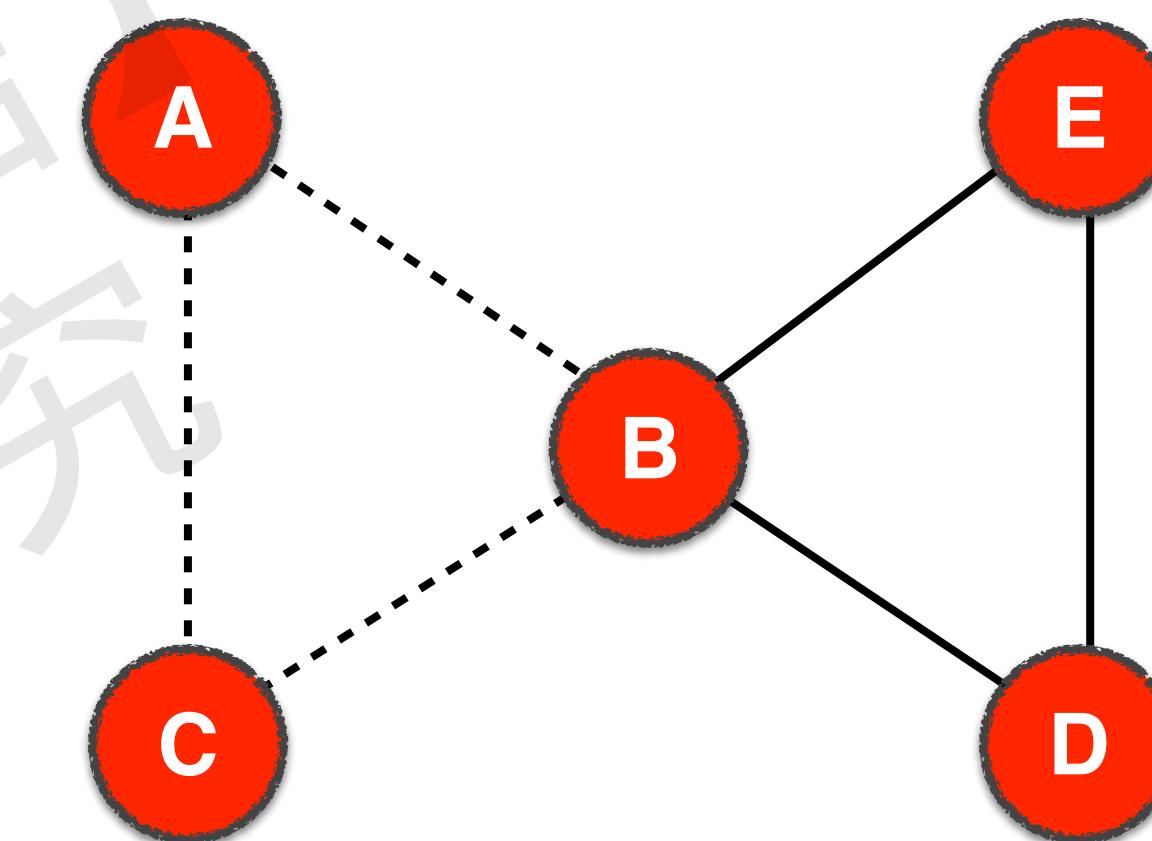
Hierholzer 算法详解

使用两个栈

curPath:



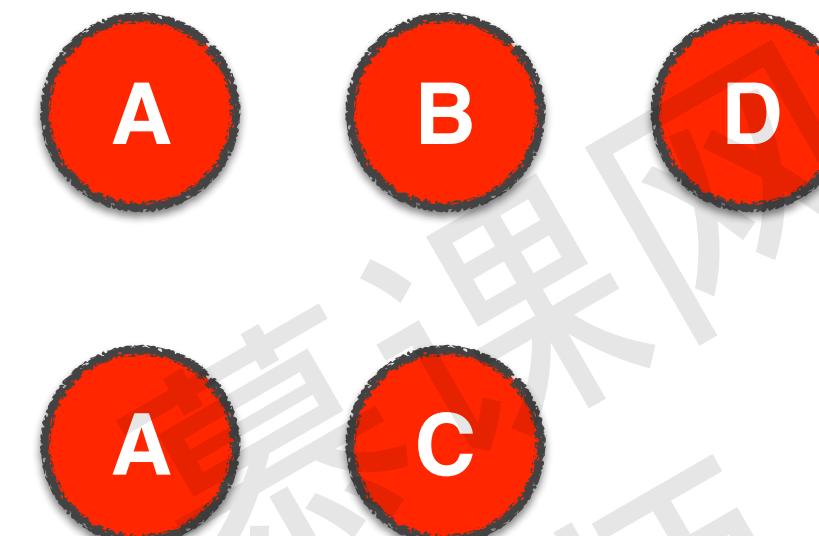
loop:



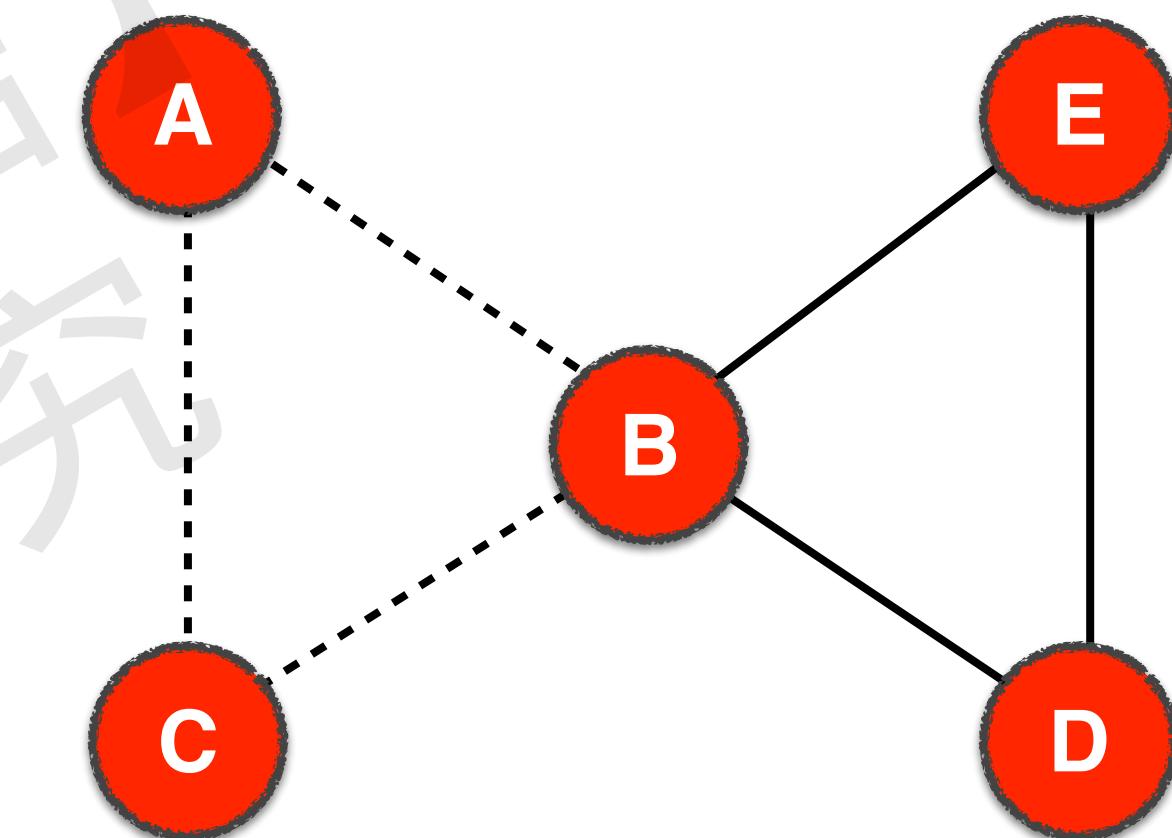
Hierholzer 算法详解

使用两个栈

curPath:



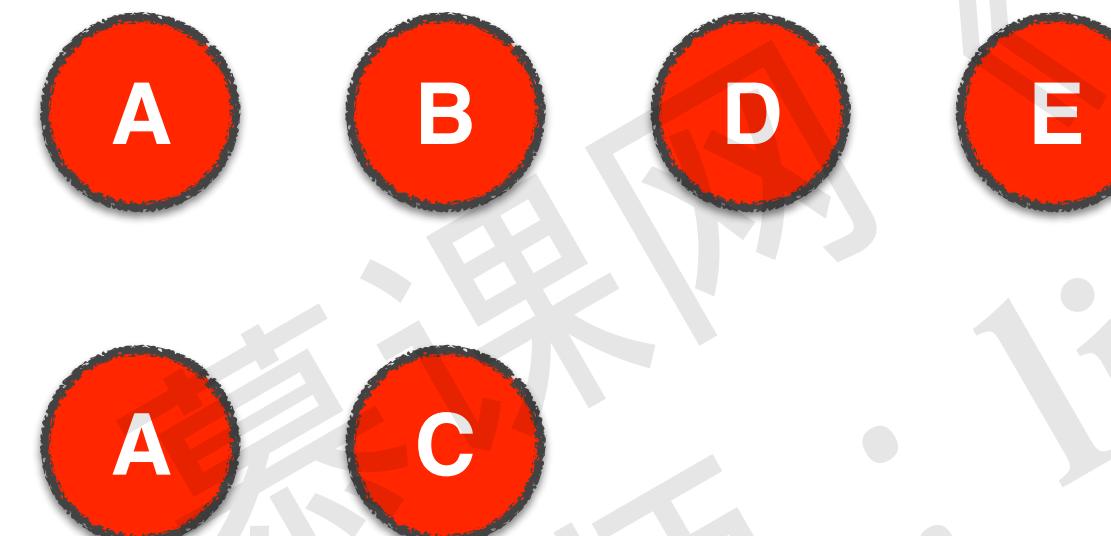
loop:



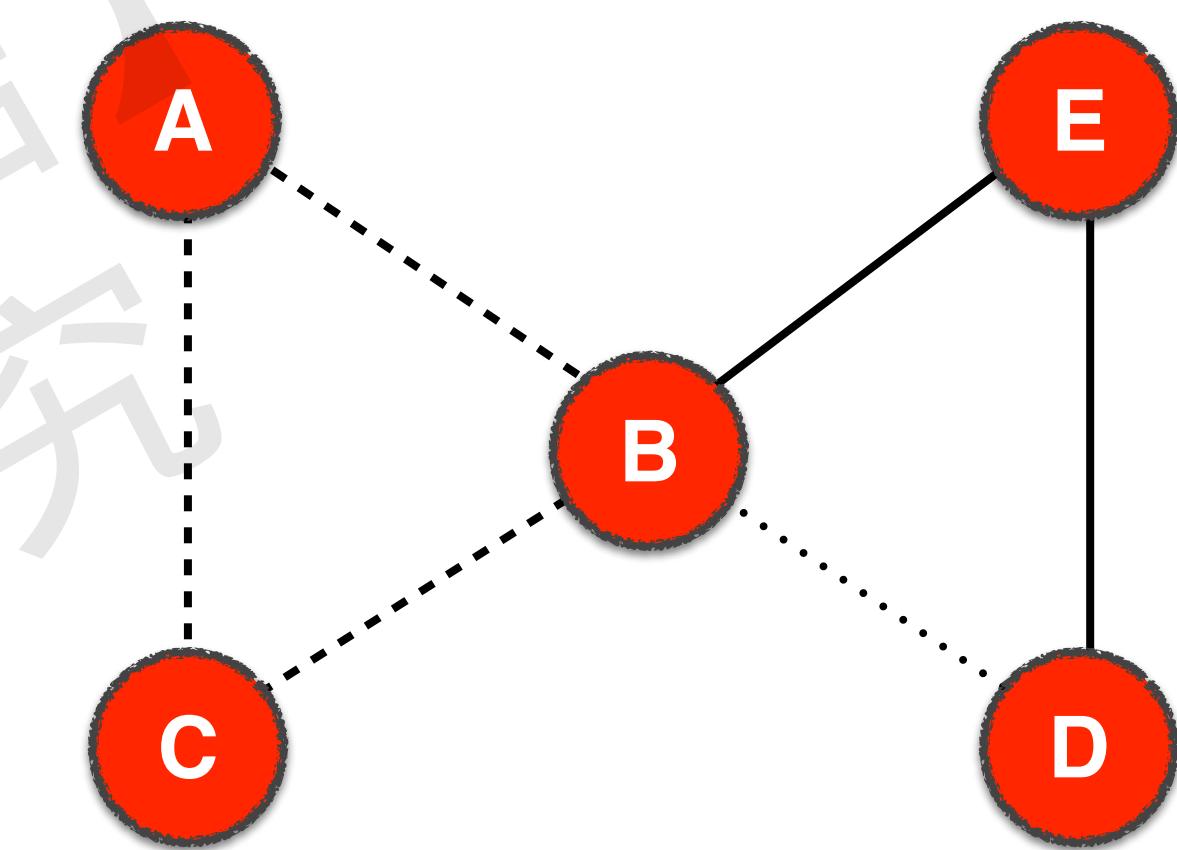
Hierholzer 算法详解

使用两个栈

curPath:



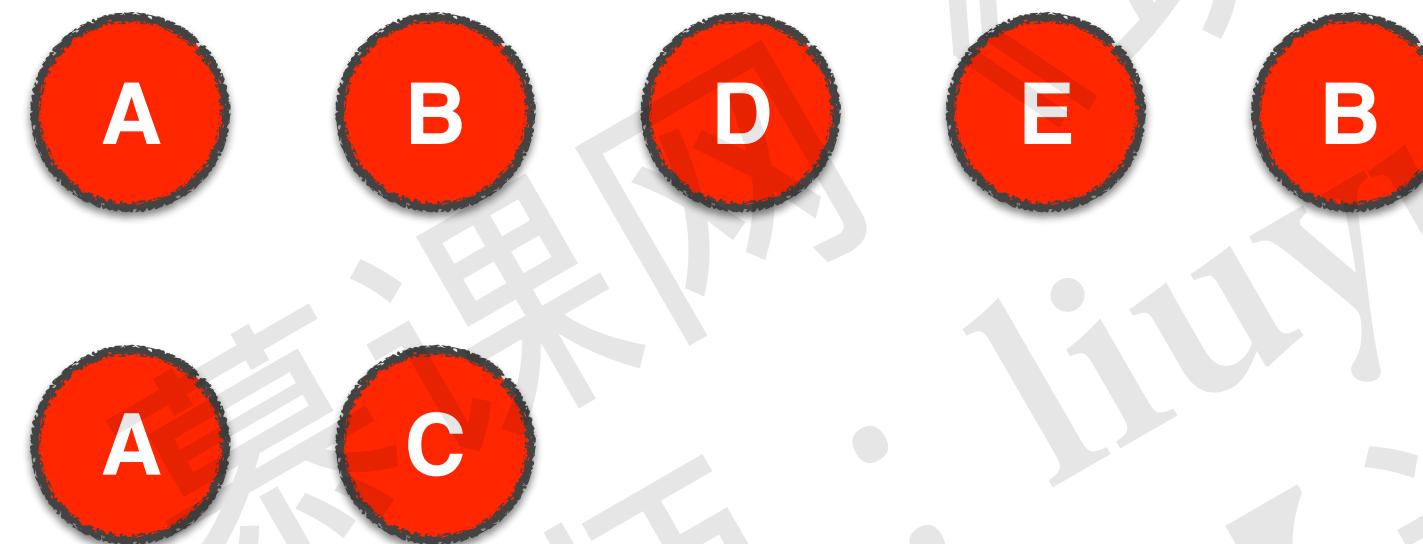
loop:



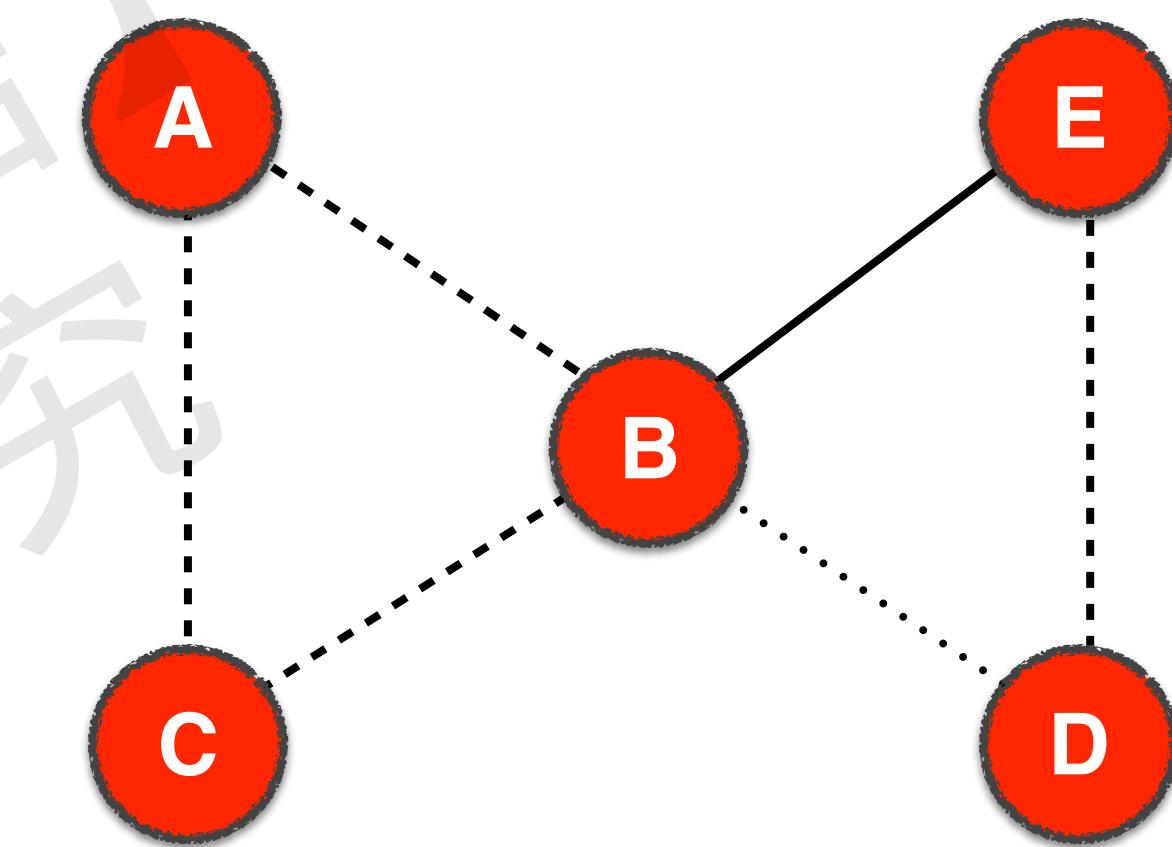
Hierholzer 算法详解

使用两个栈

curPath:



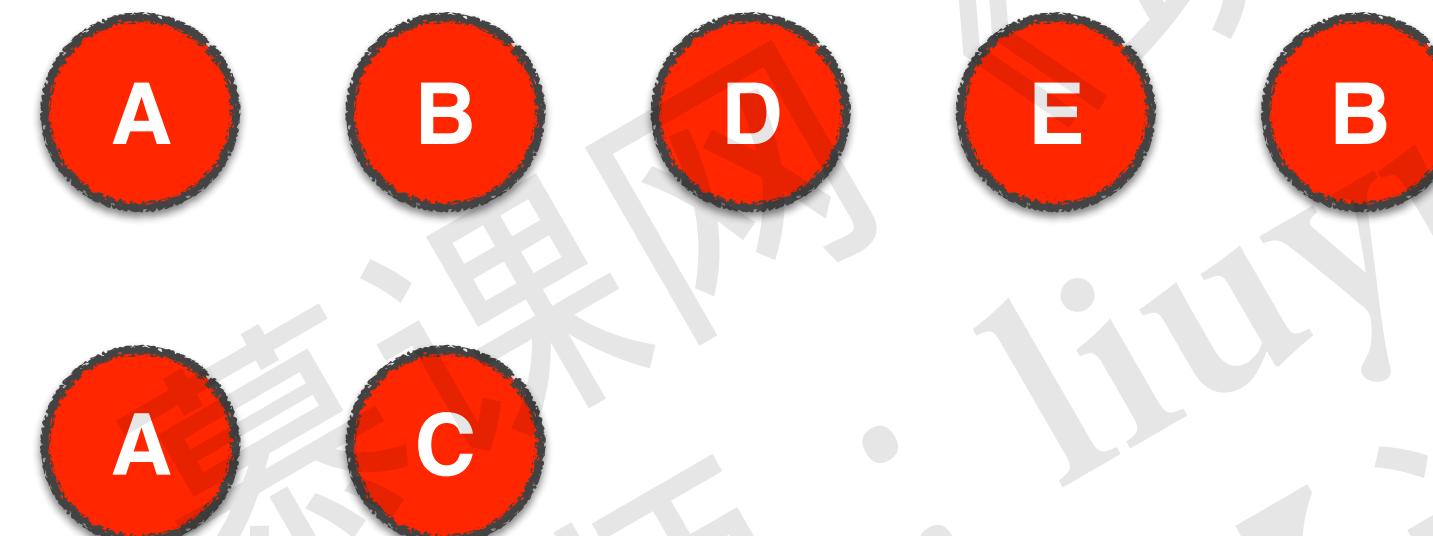
loop:



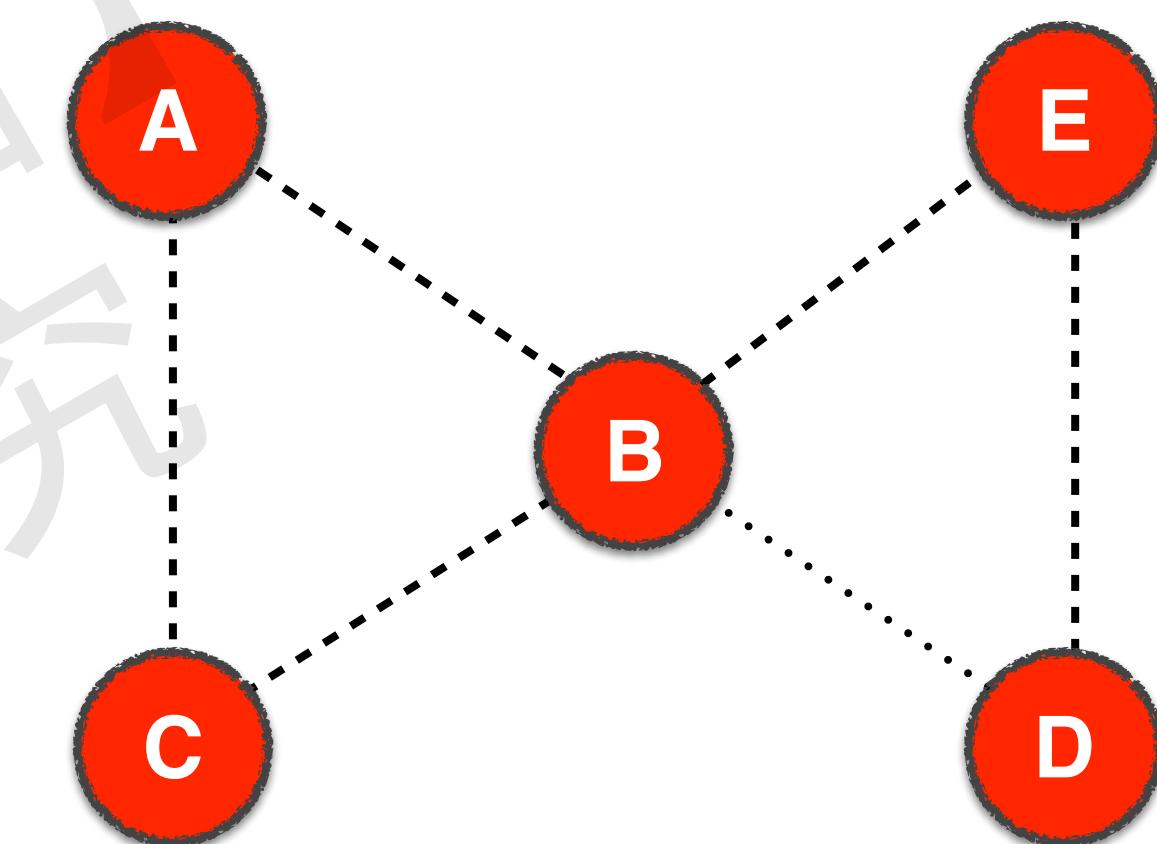
Hierholzer 算法详解

使用两个栈

curPath:



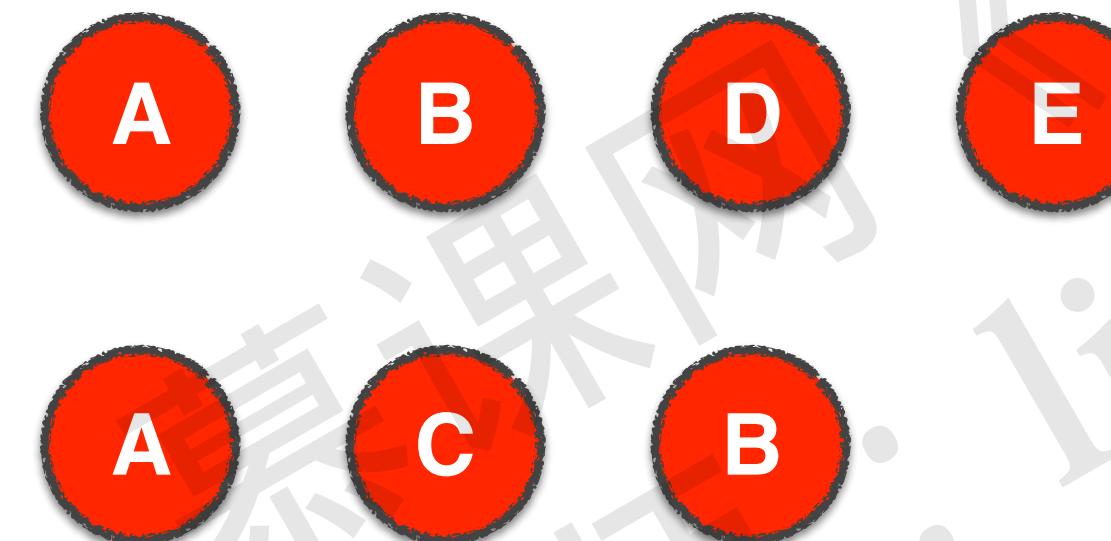
loop:



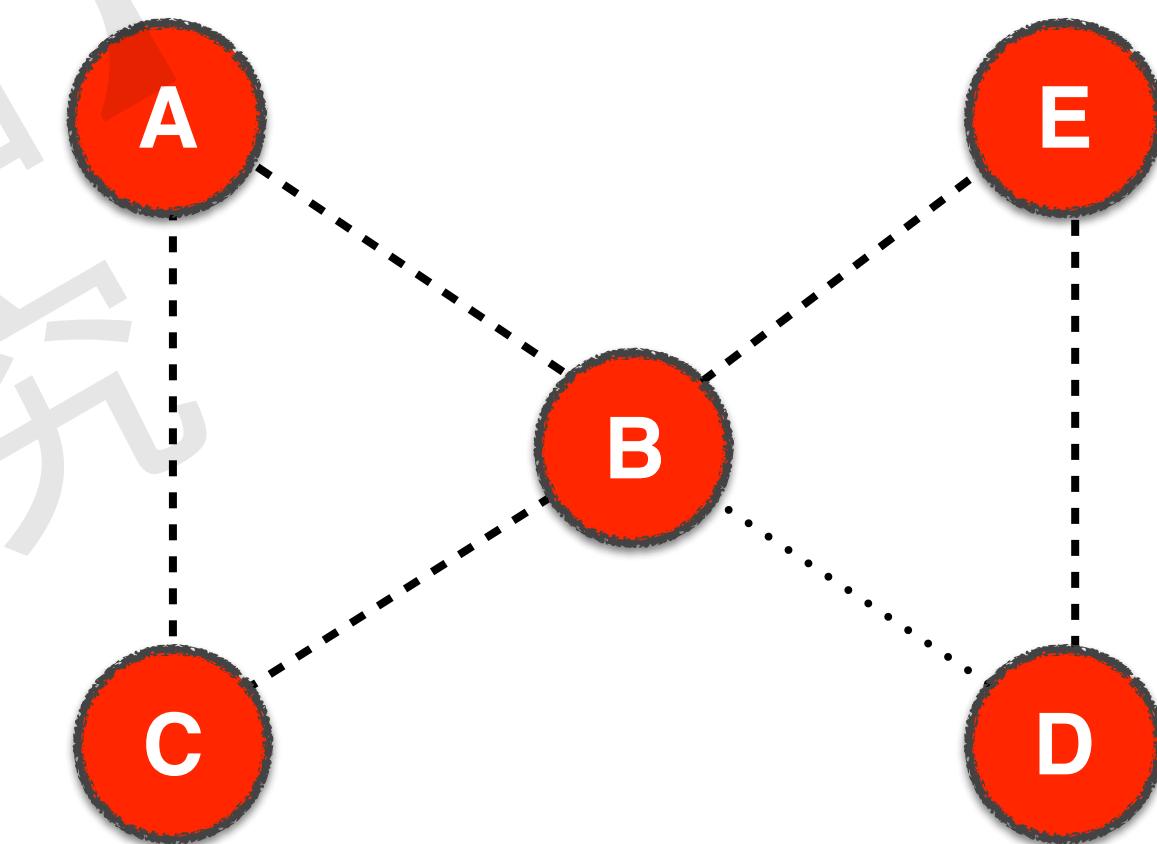
Hierholzer 算法详解

使用两个栈

curPath:



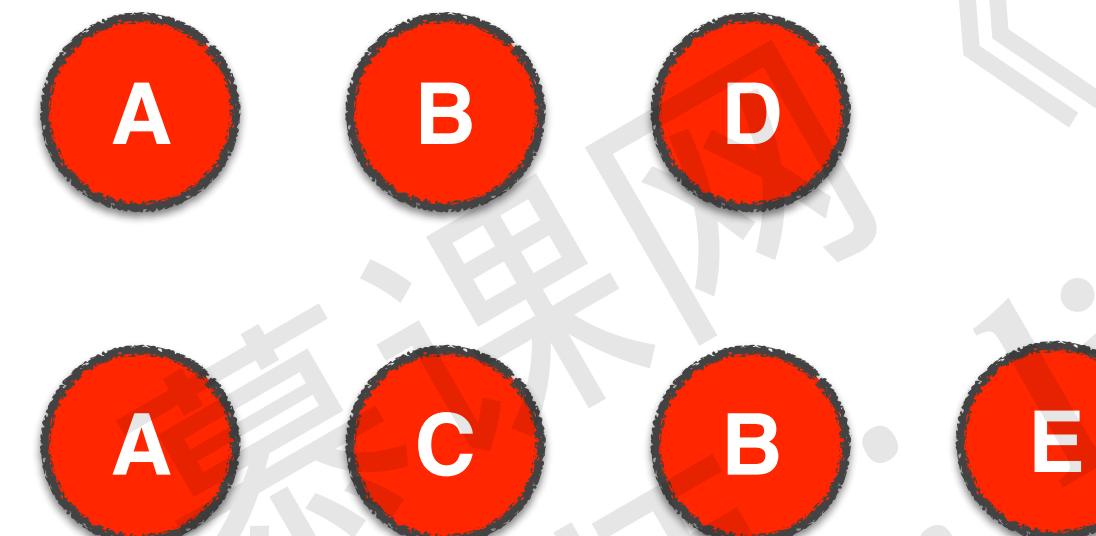
loop:



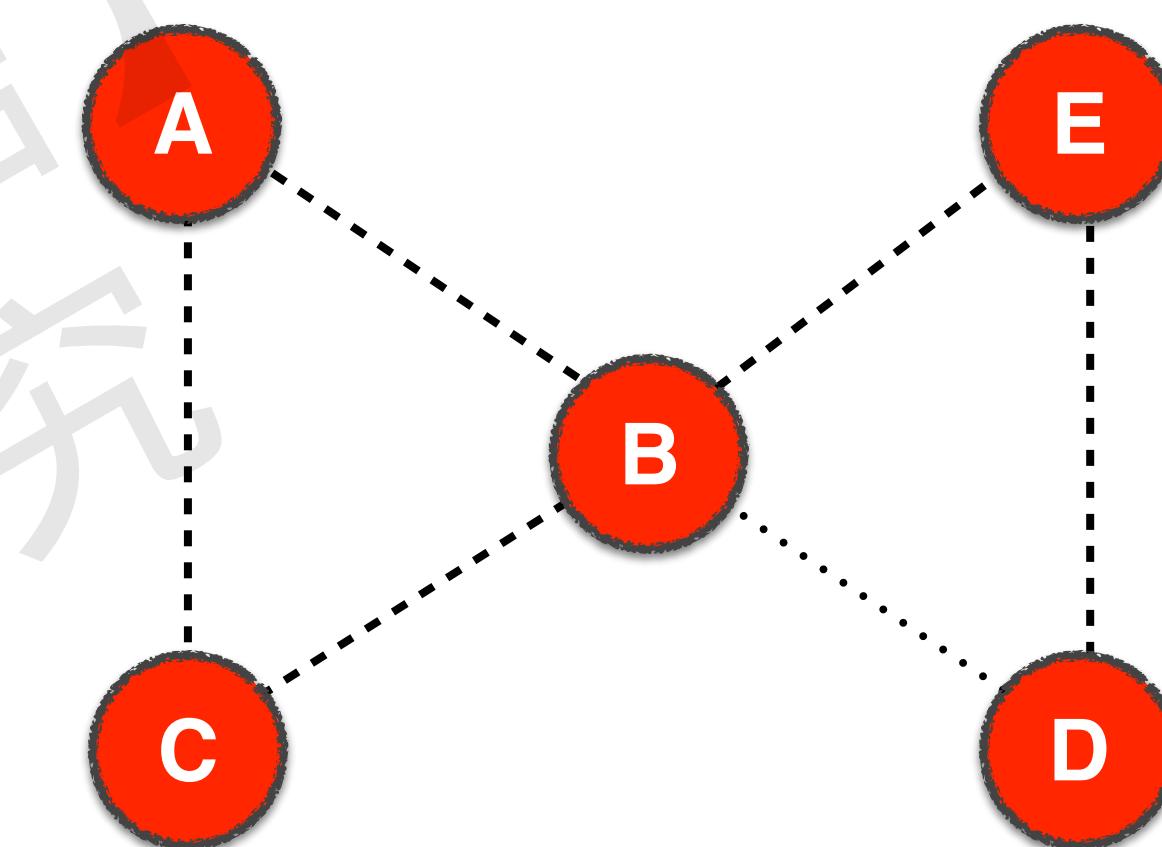
Hierholzer 算法详解

使用两个栈

curPath:



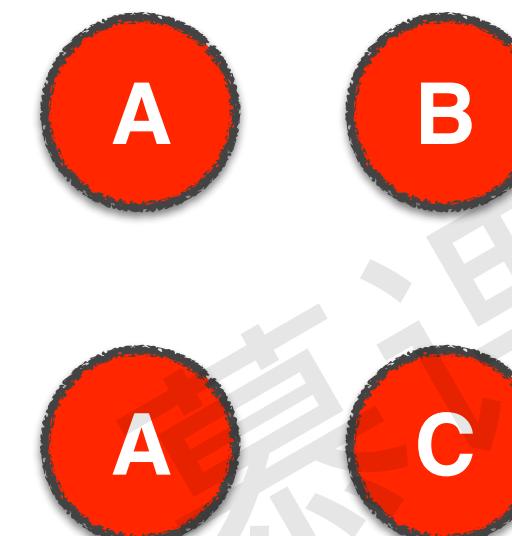
loop:



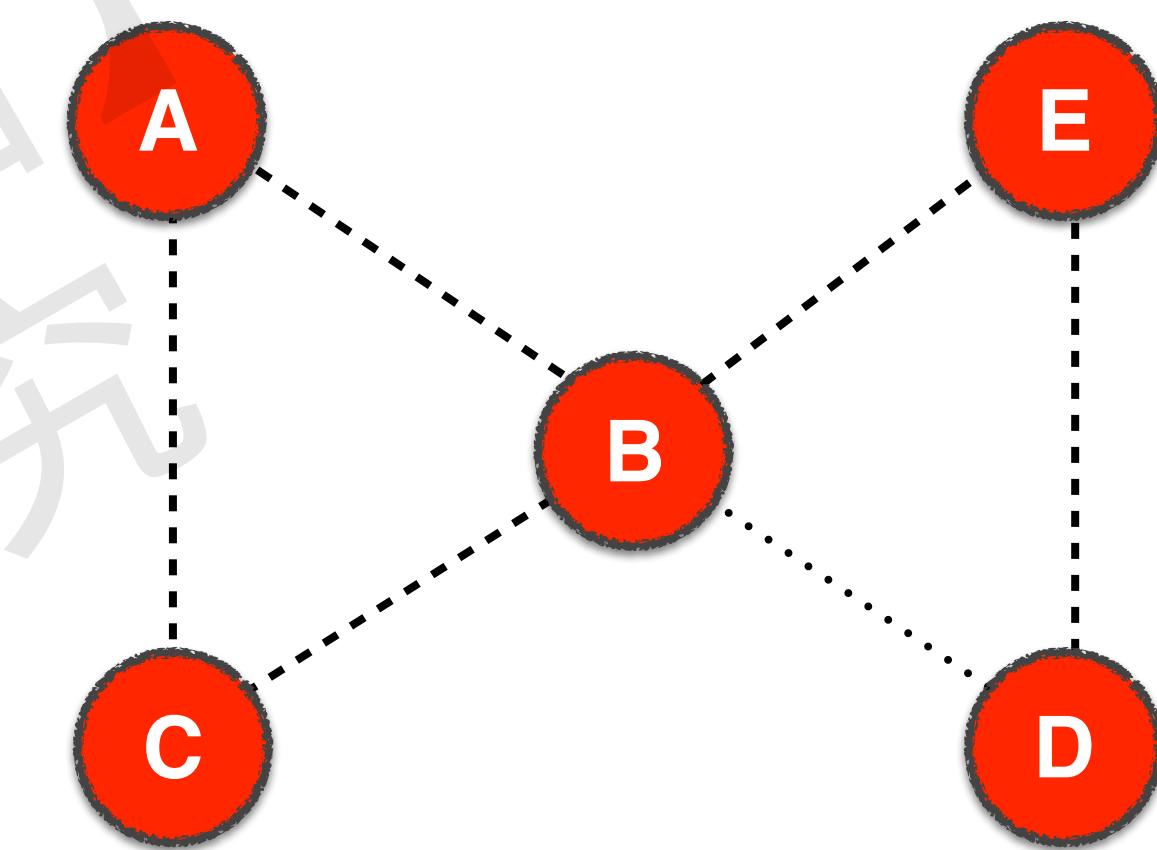
Hierholzer 算法详解

使用两个栈

curPath:



loop:



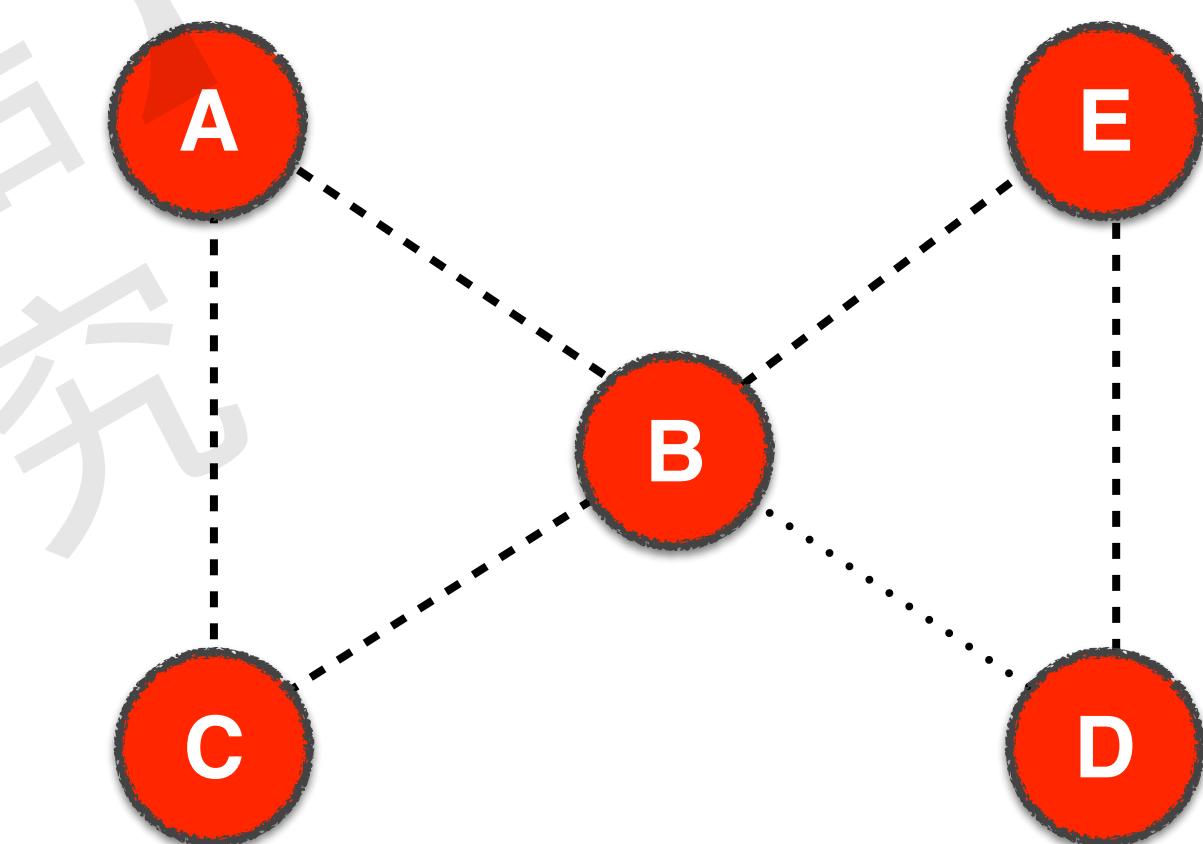
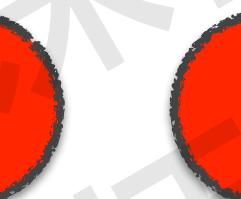
Hierholzer 算法详解

使用两个栈

curPath:



loop:



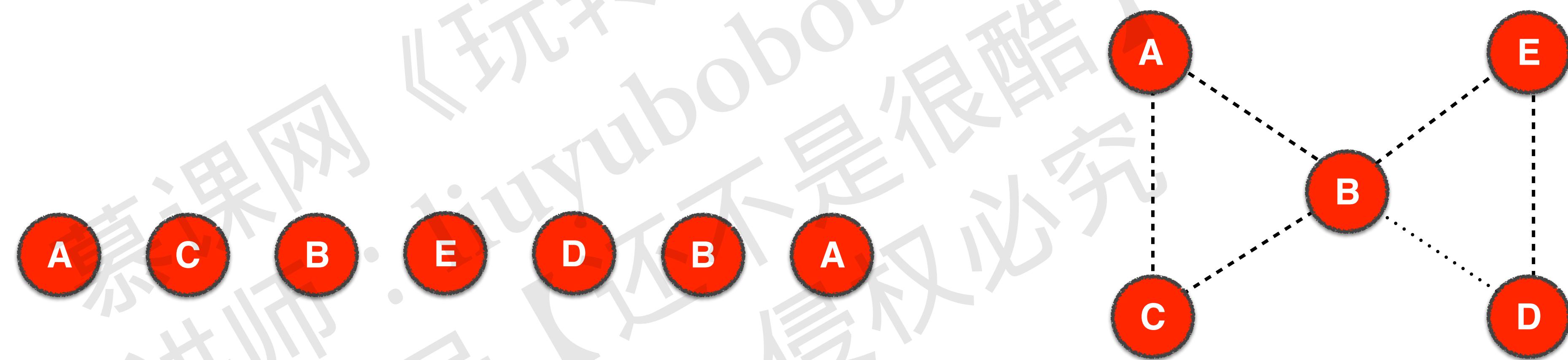
Hierholzer 算法详解

使用两个栈

curPath:

loop:

每个边走一次，回退一次 $O(E)$



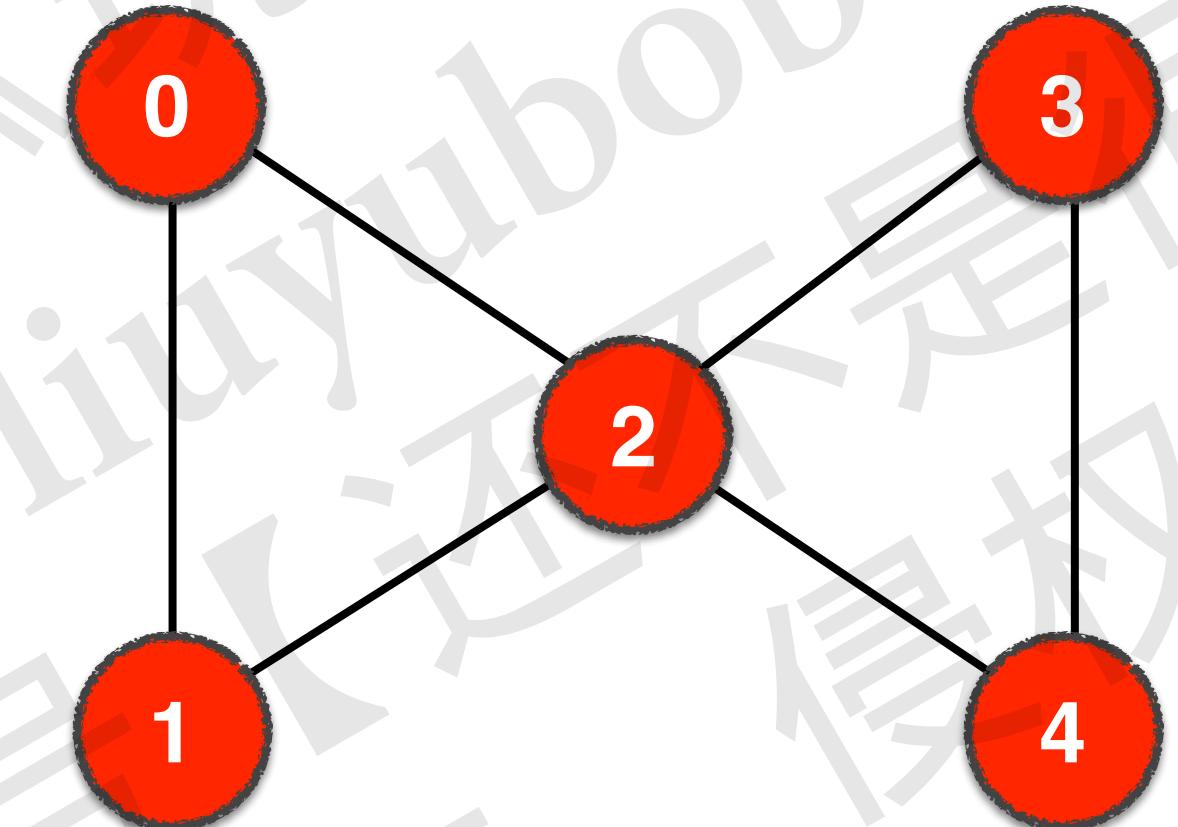
实现 Hierholzer 算法

liuyubobobo

编程实践：实现 Hierholzer 算法

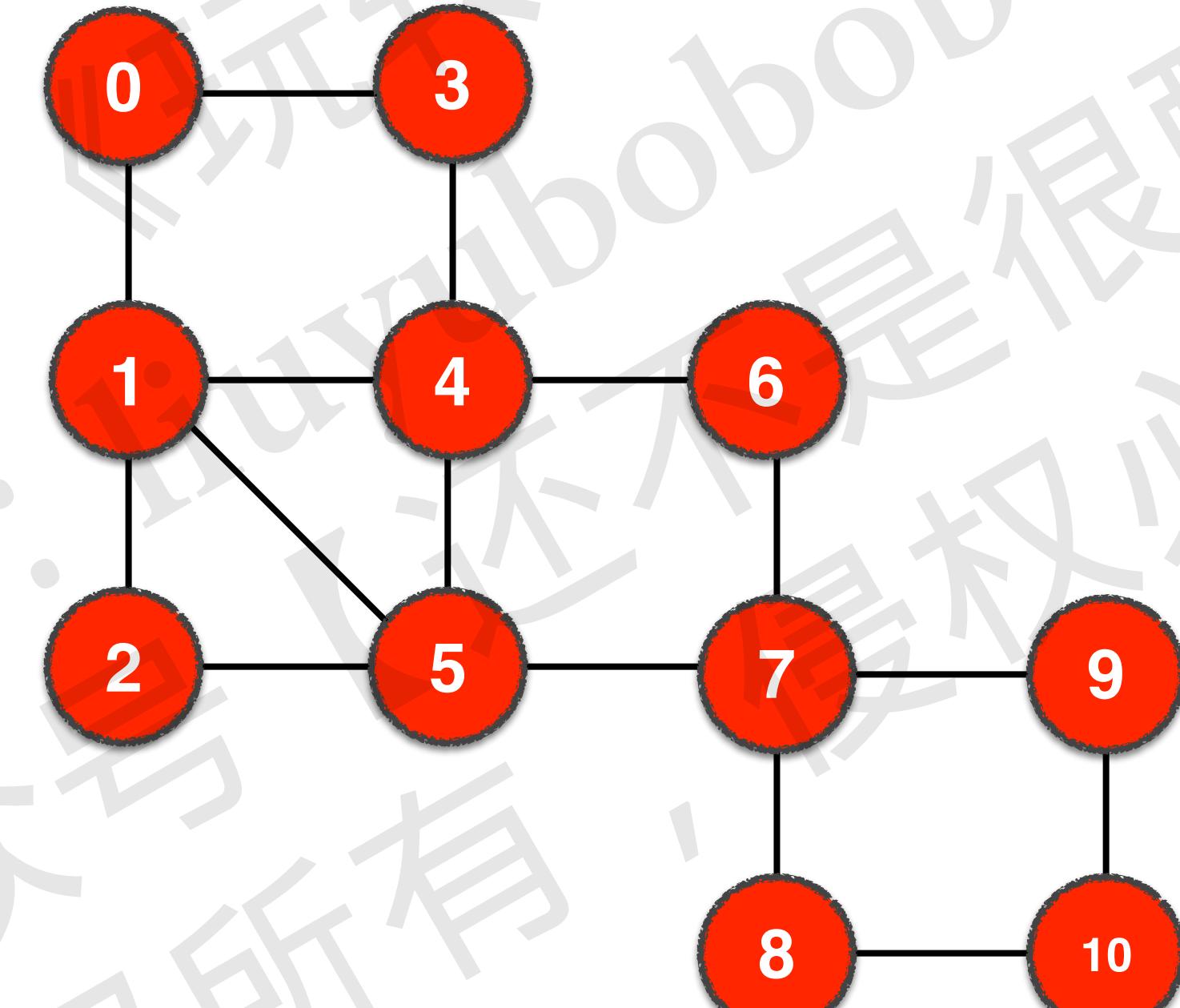
慕课网 · liuyanbo · 《玩转图论算法》
讲师：刘研波 · 版权所有，侵权必究

Hierholzer 算法详解



慕课网《玩转图论》
讲师：liuyubobobo
版权所有，侵权必究

Hierholzer 算法详解



欧拉路径和本章小结

liuyubobobo

欧拉路径

欧拉回路

从一个点出发，沿着边行走，**经过每个边恰好一次**，之后再回到出发点

欧拉路径

从一个点出发，沿着边行走，**经过每个边恰好一次**，之后到结束点
结束点和起始点可以不一样

欧拉路径

欧拉回路存在的性质

对于无向联通图

每个点的度是偶数

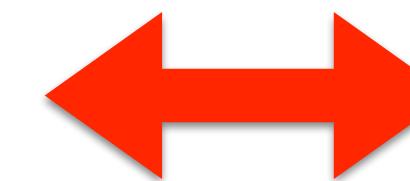


图存在欧拉回路

欧拉路径存在的性质

对于无向联通图

除了两个点每个点的度是偶数



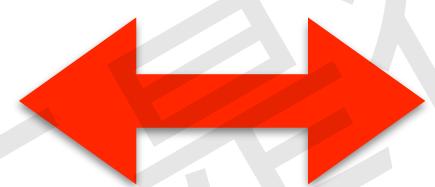
图存在欧拉路径

欧拉路径

欧拉路径存在的性质

对于无向联通图

除了两个点每个点的度是偶数



图存在欧拉路径



起始点和终止点

证明：和欧拉回路一样

算法：Hierholzer 算法

欧拉路径

实现欧拉求解路径的 Hierholzer 算法

Hierholzer 的递归算法

回溯法 和 Fleury

下一章

无向带权图

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