Kosaraju求强连通分量

主讲人：邓哲也

强连通分量的求解

求解有向图的强连通分量主要有两种算法：

Tarjan 算法：DFS 一次原图

Kosaraju算法：DFS 一次原图，DFS一次逆图

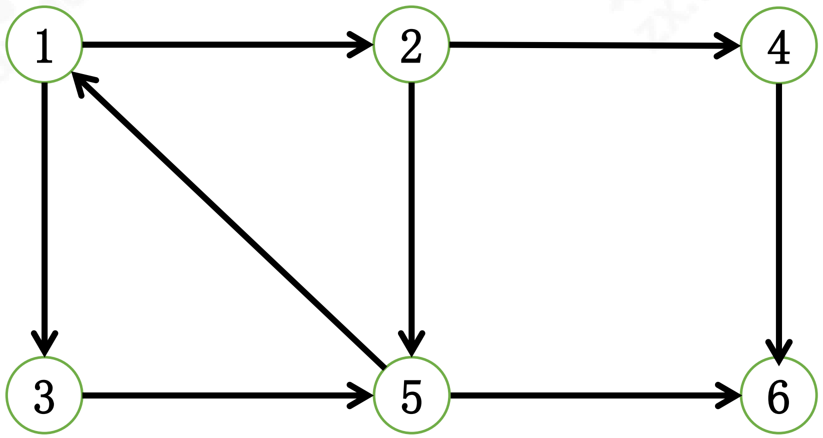
Kosaraju算法

这里我们介绍一种只需从某个顶点出发进行两次遍历：一次在原图上遍历，一次在逆图上遍历，就可以求出图中所有强连通分量的 Kosaraju 算法。

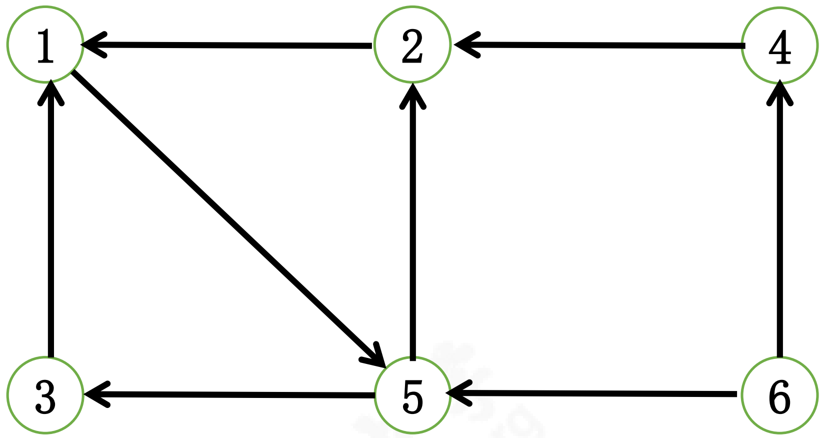
时间复杂度：O(n + m)

逆图

对于一个有向图 G，把 G 中的所有边都反向，就得到了 G 的逆图 G’。



原图 G



逆图 G’

Kosaraju算法的思想

如果有向图 G 的一个子图 K 是一个强连通子图。

那么各边反向之后，不影响 K 的强连通性。

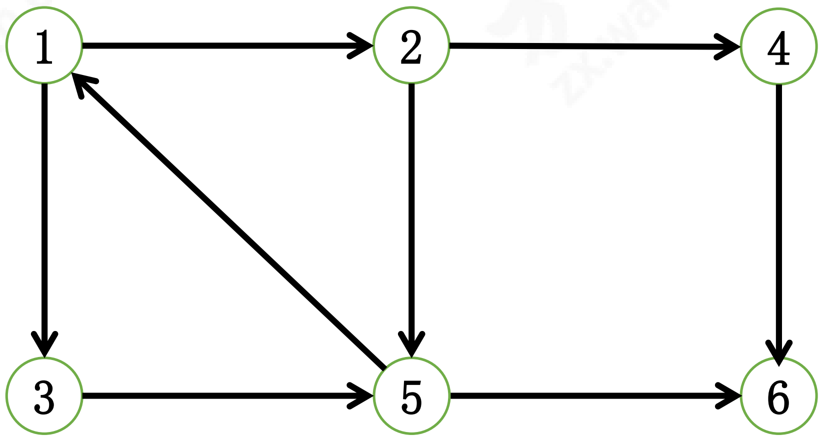
但如果 K 是单向连通的，各边反向后可能某些顶点间就不连通了。

Kosaraju算法的过程

1. 对原图 G 进行 DFS，记录每个节点 DFS 结束的时间 dfn。
2. 将 G 中的每条边反向，得到逆图 G’。
3. 选择当前 dfn 最大的顶点出发，对 G’ 进行 DFS，删除能够遍历到的点，这些点构成一个强连通分量。
4. 如果还有顶点没有删除，就继续执行第 3 步，否则算法结束。

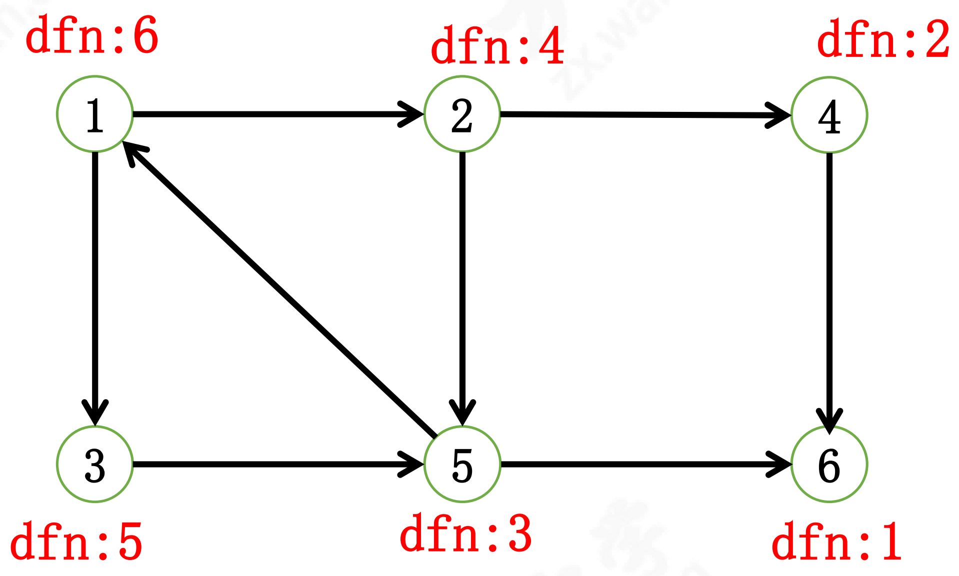
Kosaraju算法

以这张图为例，我们来看看 Kosaraju 算法是如何运行的。



Kosaraju算法

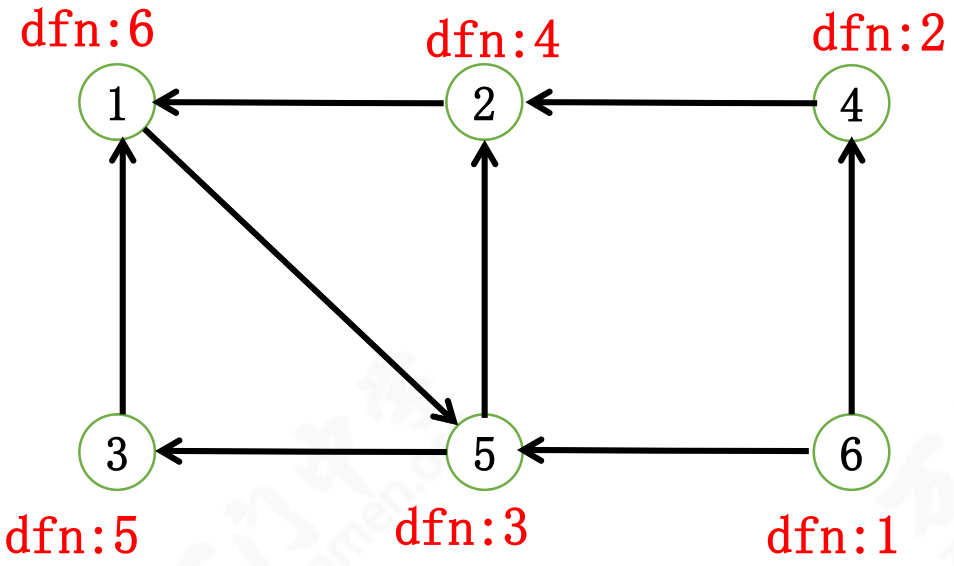
首先在 G 上 DFS，记录每个点的 dfn。



Kosaraju算法

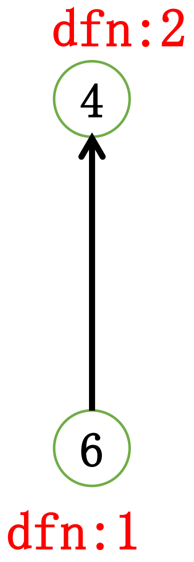
对逆图 G’ 进行 DFS，每次从 dfn 最大的开始 dfs。

这里从 1 开始 DFS，得到 {1,2,3,5}



Kosaraju算法

从 4 开始 DFS，得到 {4}



Kosaraju算法

从 6 开始 DFS，得到 {6}

Kosaraju算法

因此只需要通过两遍 dfs 就可以求出所有的强连通分量：

{1, 2, 3, 5}, {4}, {6}.

时间复杂度为 $O(n + m)$

下节课再见