Graph Algorithm V

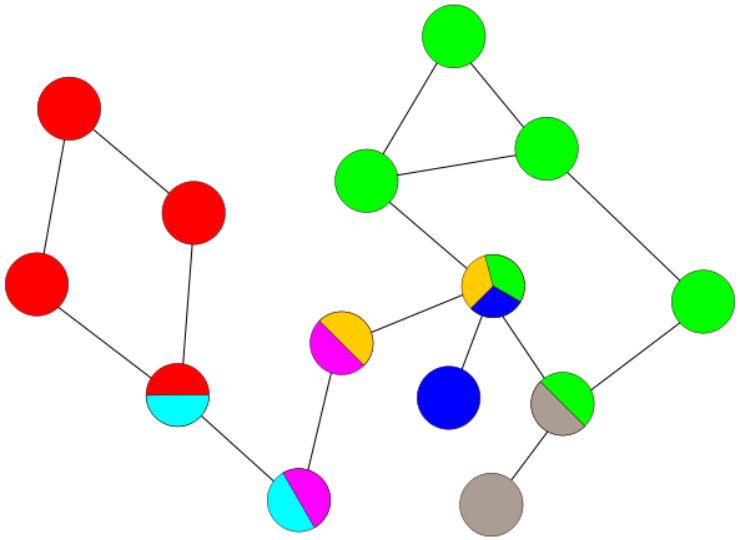
Instructor: Shizhe Zhou

Course Code:00125401

Biconnected-Component

- Menger定理:[关于有限无向图的连通性的基本定理]

令 $G=(V,E)$是一个无向连通图，并令 u 和 v 是 G 中两个非邻接顶点。为了把 u 和 v 分离，所需从 G 中删除的最少顶点数等于从 u 到 v 之间不相交的路径的最大数。（当一个顶点被删除时，所有与之相连的边也同时被删除。）



terminology

- 链、迹、路是图论中上个相似的概念，分别如下:

1、链（chain or walk）：顶点和边交错出现的序列称为链，在序列中边的前后两个顶点正好是边的端点，序列的第一个顶点和最后一个顶点为链的端点，其余的点为内点。
2、迹（trail）：边互不相同的链称为迹。即迹中无重边。
3、路（path）：内部点互不相同的链称为路。即路中无重点。

闭链（迹、路）：两端点相同的链（迹、路）称为闭链（迹、路）。

从上面的定义知道，三者是有区别的，迹、路也是链，但链不一定是迹、路。同样的闭链、闭迹、闭路也是有相同和区别的。

闭合的链==闭链**[本书中闭链==闭路]**

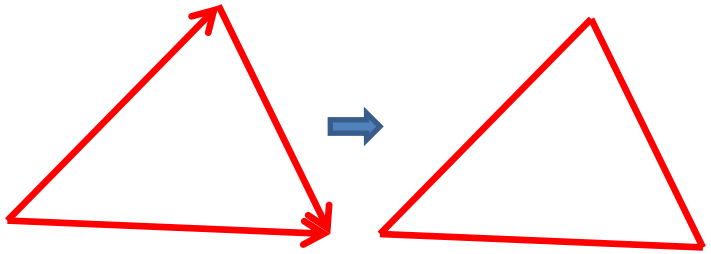
4、连通的（connected）:存在连接x到y的路，则称x到y是连通的。
因为链中肯定存在路，所以闭链是连通的，但不能说是环路，不过反过来说就可以了，即环路是连通的闭链

无向图的连通性

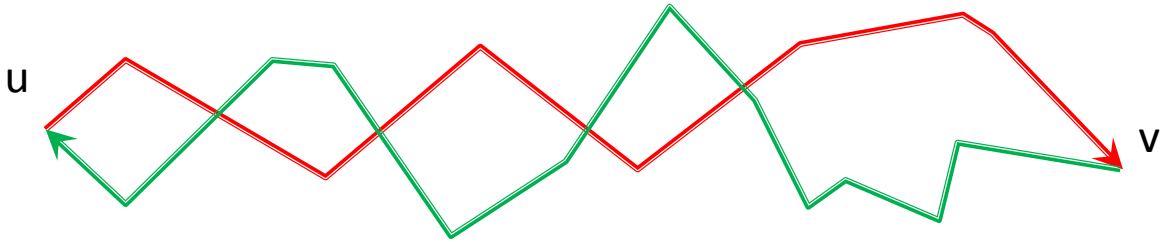
- 对无向图G=(V,E)，其任意一对顶点u和v间存在一条路径相连通，则称G是连通的.

有向图的连通性

- 对有向图G=(V,E).
- 它的Undirected form



- 如果有向G的Undirected form中任何两个顶点有路径，则称G为弱连通的.
- 如果有向G中任何两个顶点u,v, 存在u到v的有向路径或者v到u有向的路径，则称G为连通的. path(u,v)||path(v,u)
- 如果有向G中任2个顶点的对(u, v)和(v, u)都存在路径相连，则称G为强连通的. path(u,v)&&path(v,u)



- 有向图G是强连通图，等价于G中存在一条回路包含所有节点至少一次.
- 无回路的连通无向图等价于树
- 下面先讨论无向图

K-连通图

- 图G满足如下:

1. #V>k;
2. 如果从G中删除任意<k个顶点及与它们相连的边, 图仍然连通, 则称G是K-vertex-connected或者K-connected.

性质：条件2等价于→”至少要删除k个顶点才能使得G不连通”.

K-连通图常见实例:
1-连通图: 连通图 <connected>
2-连通图: 双连通图 <biconnected>
3-连通图: 三连通图 <triconnected>

2连通图(双连通图)

定义：对连通图G满足如下条件:删除任意一个节点,图的剩余部分仍然连通.

性质：2连通图，没有所谓”关节点”.

关节点: 如果删除v和v的边，图的连通分量数目增加了(即图分离了)，那么v是关节点。

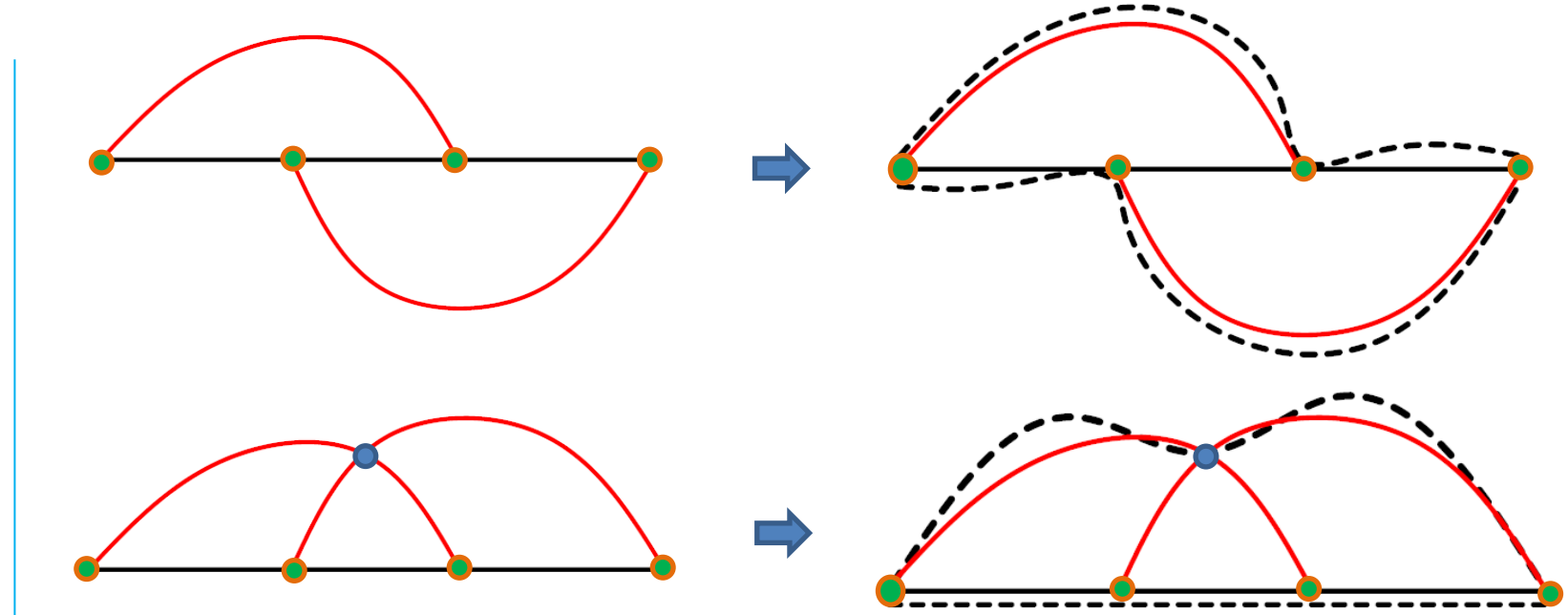
等价性：2连通图等价于没有关节点.

- 证明:　即证明对任意顶点v, 删除v以及与v相连的边之后, G仍然保持连通.

[必要性]　　对任意点对(a,b), 由2连通性, 存在两条分离路径p1, p2; 如果v在p1上, 则v必不在p2上;或者v即不在p1也不在p2上, 删除v及其所有的边最多使得p1或p2中的一条被打断, 因此a和b仍然保持连通.

[充分性]　因为没有关节点, 故任意一个顶点的一邻域中的所有点对都是2连通的(因为删除关节点同时需要删除其所有边). 可由传递性(见下页图)推出不属于同一点的一邻域的其他所有点对间也存在至少两条路径相连。

Th1：通过边数为3的路径相连的任意两点间存在两条分离路径.
证明如下图所示:



Th2 : 通过边数为4的路径相连的任意两点间存在两条分离路径 .
可基于Th1的结论，由同理推出.

使用Th1为基础，可归纳证得：
Th3 : 通过任意边数的路径相连的两点间存在两条分离路径 .

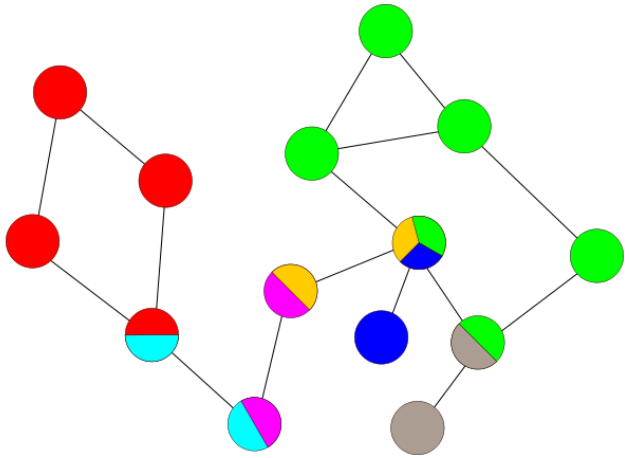
2连通图的性质*

- e, f是2连通图G的两条边,则存在一个包含着e和f的闭链.
- 证明: 设e=e(v,w),f=(s,t)由2连通性, 知存在路径e'(v,w),e'不包含e. 对G作如下修改:增加节点z, 增加边(v,z)和(w,z),删除边e. 再类似的对f增加点y并连接两边, 得到G'. 现证明G'仍然是2连通的. 由等价性,只需证明G'不含关节点. 删除z和边(v,z),(w,z)得到的图等于G-e. 在G中所有使用了边e的路径都可以绕行e', 因此所有点对之间在G-e中仍然有路径连通，即G-e是连通图. 因此z不是关节点. 同理可证明y不是G' 的关节点. e, f的四个端点v,w,s,t也不可能是G' 的关节点(because 这四点在原G中被删除后就不可能打破连通性, 即剩余部分连通,在G'中剩余部分是原来G的剩余部分连出来一条边，自然也是连通的); G'中剩余的其他所有点的连接关系和在G中一样, 也不可能是关节点. 因此G'是2连通图, 所以z,y间存在两条分离路径, 不妨设为p1(z,v...s,y)和p2(z,w...t,y);现从p1和p2上取出修改G'新加入的四条边, 得到分离的两条路径p1'(v...s)和p2'(w...t), 他们和e与f一起构成闭链.

双连通分量==最大连通子图

- 一个边的最大子集, 其导出子图是双连通的. 即不存在一个更大的包含它在内的边子集, 其导出子图也是双连通的.
- 双连通分量对G的边集合作了一个划分，但是顶点可以属于多个双连通分支。这样的顶点可以证明就是关节点.(删除之使得连通分量数增加).

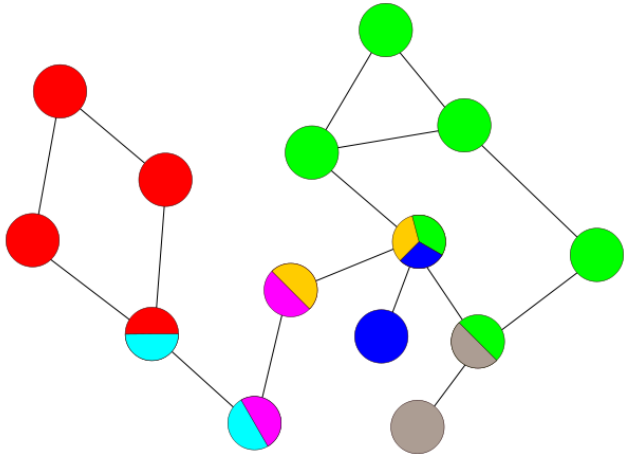
[*一个双连通分支可以仅包含一条边*]



Biconnected-Component

- Menger定理:[关于有限无向图的连通性的基本定理]

令 $G=(V,E)$是一个无向连通图，并令 u 和 v 是 G 中两个非邻接顶点。为了把 u 和 v 分离，所需从 G 中删除的最少顶点数等于从 u 到 v 之间不相交的路径的最大数。（当一个顶点被删除时，所有与之相连的边也同时被删除。）



All hamiltonian graph is biconnected

双连通分支划分,唯一存在

- 引理#:一个双连通分支完整的包含G中的某一个闭链.(仅对边数>=2的分支).

证明: 如不完整包含, 则该闭链的剩余部分可加入该分支, 且加入的顶点没有关节点.这与分支的最大性矛盾.

→: 2个双连通分支**最多**相切于一个关节点.

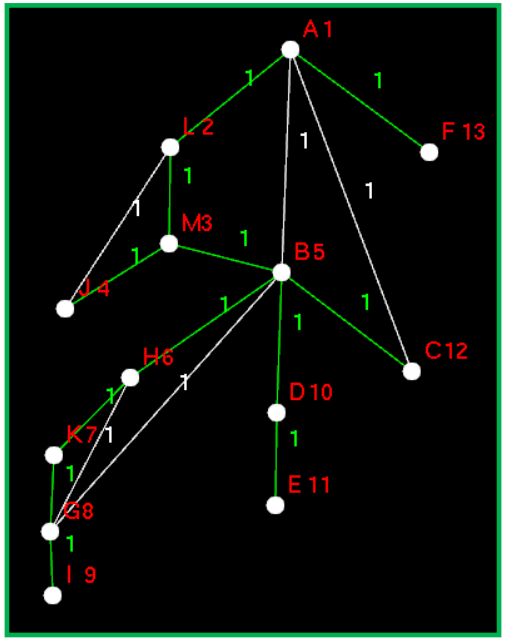
- 结合引理#和性质*, 即有[引理7.9]:

如果边e和f属于同一个双连通分支, 则在该分支中存在一个闭链包含e和f; 如果在G中存在一个闭链包含G的两条边e和f, 由于该闭链只能完整的属于一个分支(note:可能存在多条闭链包含e和f)，因此e和f也只能属于同一个分支.

7.9直接导出7.10:

一条边属于且仅属于一个双连通分支.

计算双连通分支



- 计算双连通分支，核心是寻找关节点.

利用DFS树的"隔离"功能: 不能横穿，只能后退!

利用DFS树寻找关节点:

- 树的根如果有多颗子树，因为G中没有横穿DFS的边，则这些子树对应的分支必然独立. 因此这样的根是关节点.

DFS在跨越分支前(扫描新边或者回朔)，必须首先回到关节点.

- 树中其余顶点v: 如果v的*某*子树中没有回边指向v的任何一个祖先节点, 则v为关节点.

(准确来说,v是相对于该子树的关节点, 因为如果删除v, 这棵子树将从图上分离.)

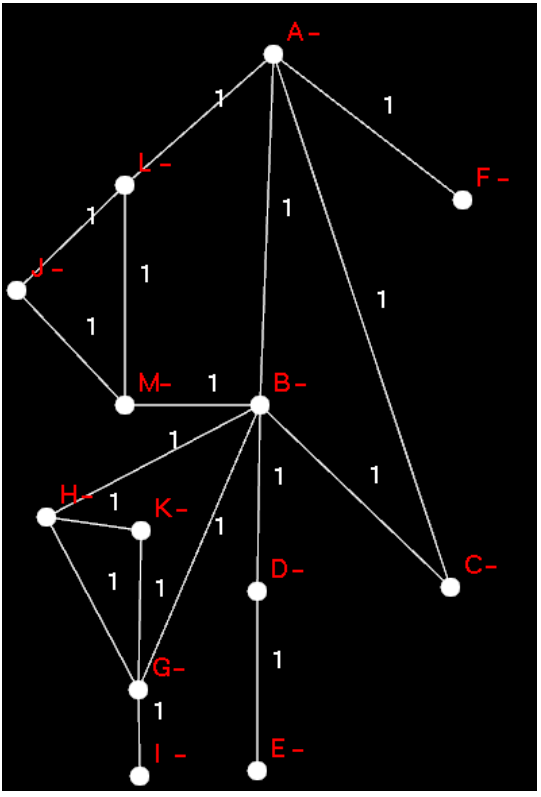
算法

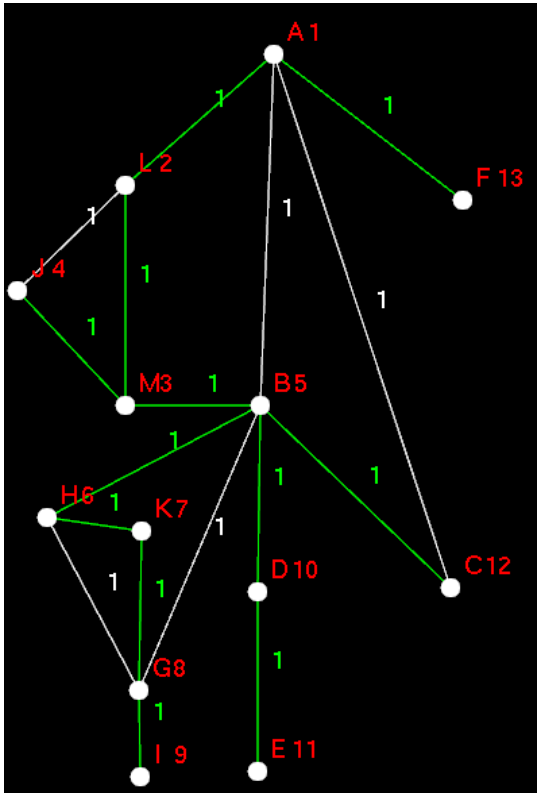
- 嵌入在一次DFS过程中, 不断的查询当前顶点是否拥有一棵子树:这颗子树没有越过该点的后退边.
- 定义high值, high(v)表示v点及其(以v的某个dfs儿子w为根的)子树发出的后退边(含指向父亲节点的树边)连接到的最高顶点的DFS number(代码中就是order).

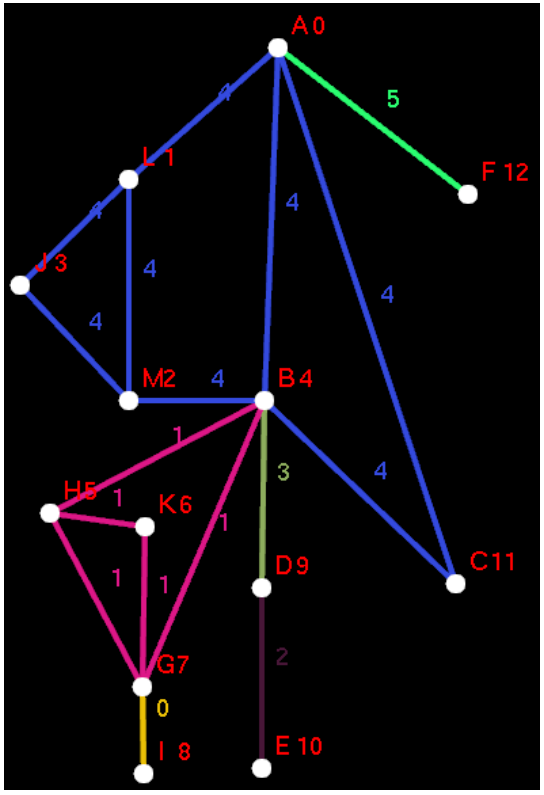
$$high(v) = min(high(w), order(k));$$

w是v在DFS树上的孩子节点，k是v在DFS树上由回边连接到的祖先节点.

high值越大,说明越在底层的连通分支；high值越小,在DFS树中的位置越高.

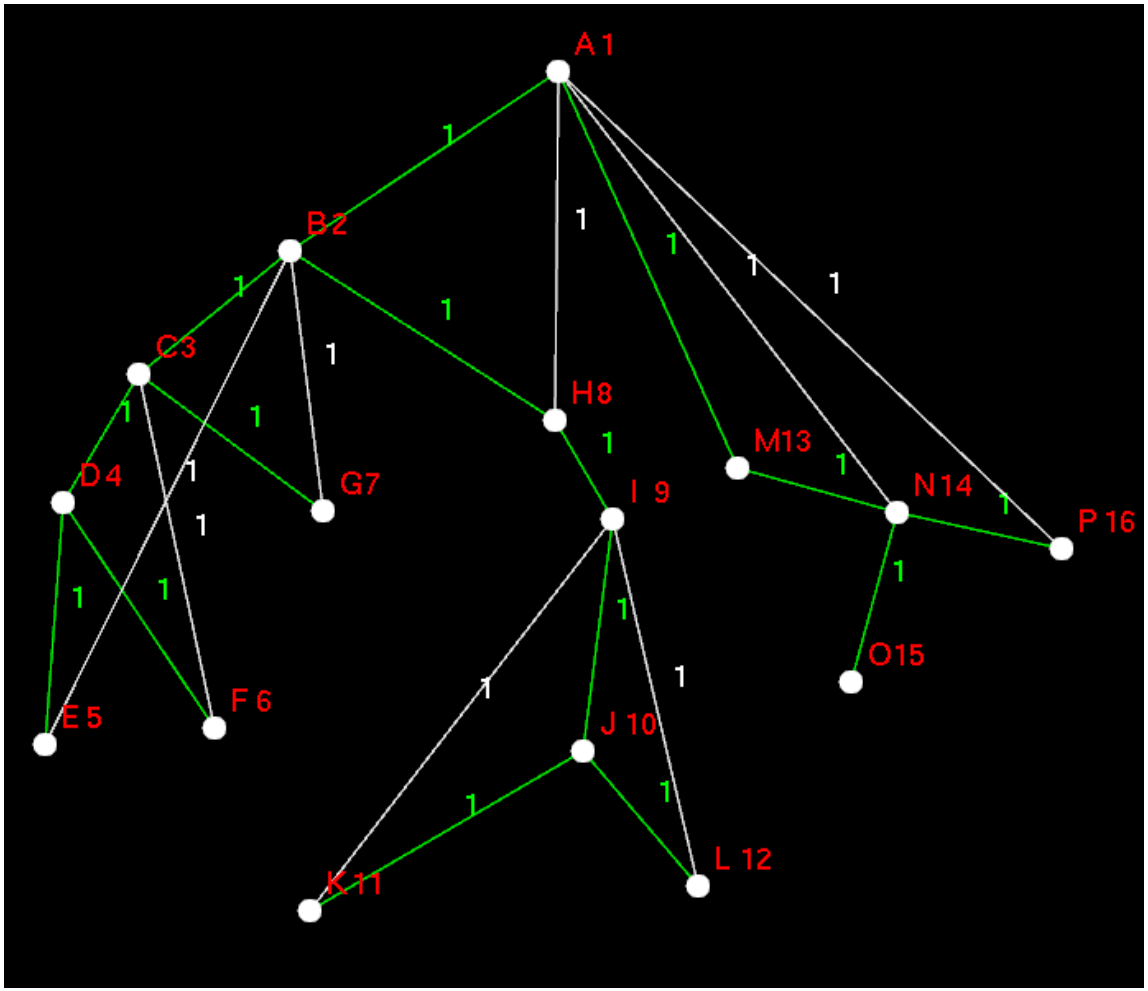




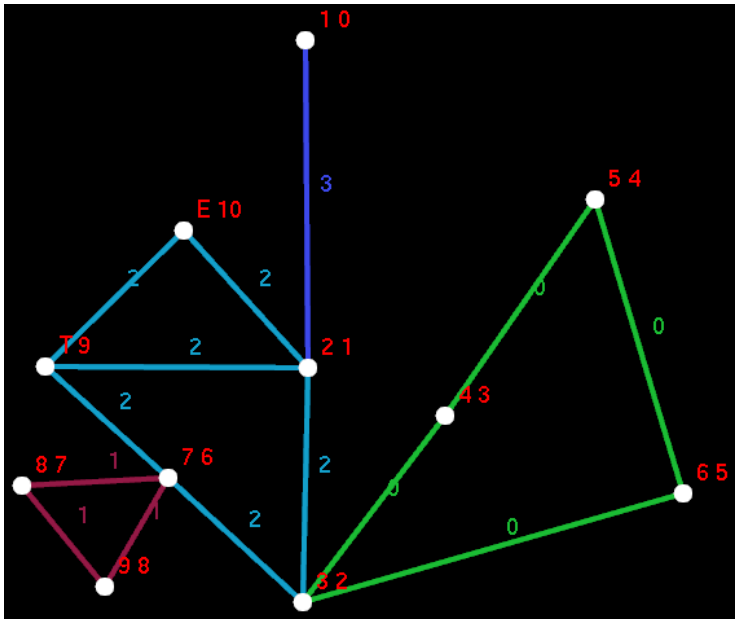


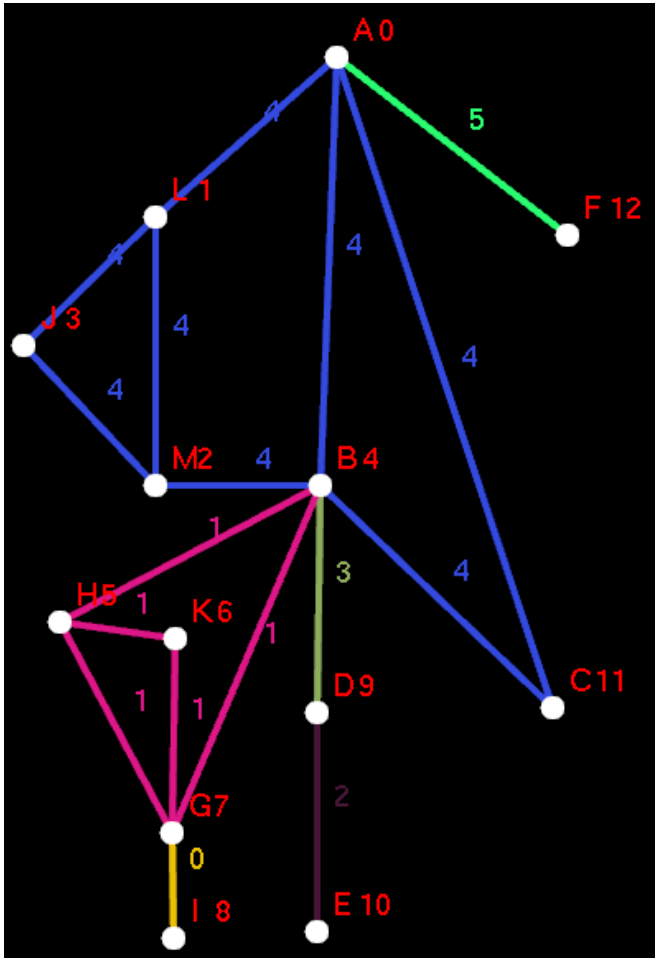

```
[high value]:
A_0
B_0
C_0
D_9
E_10
F_12
G_4
H_4
I_8
J_1
K_4
L_0
M_0
```

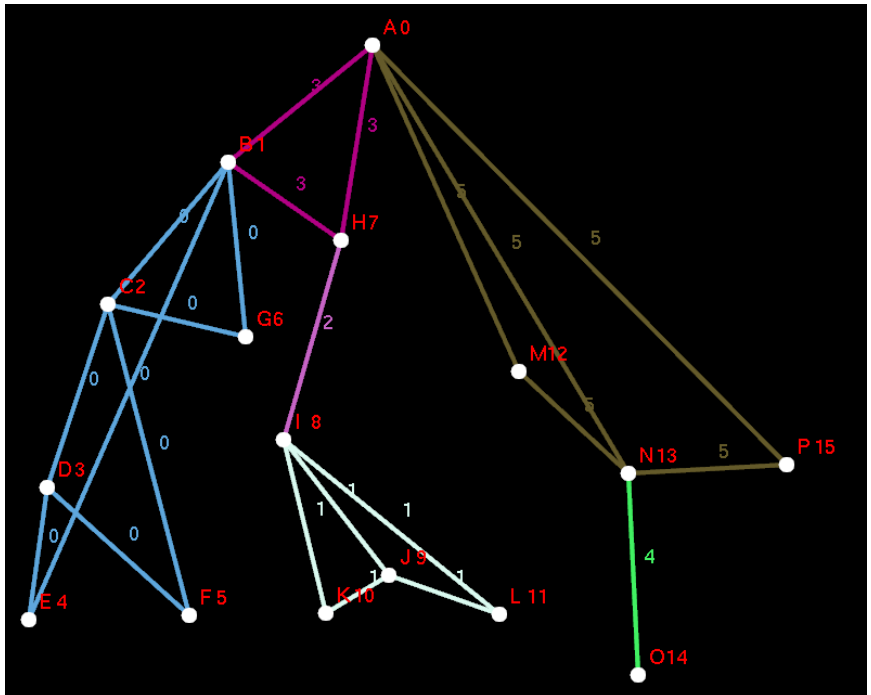
DFS tree



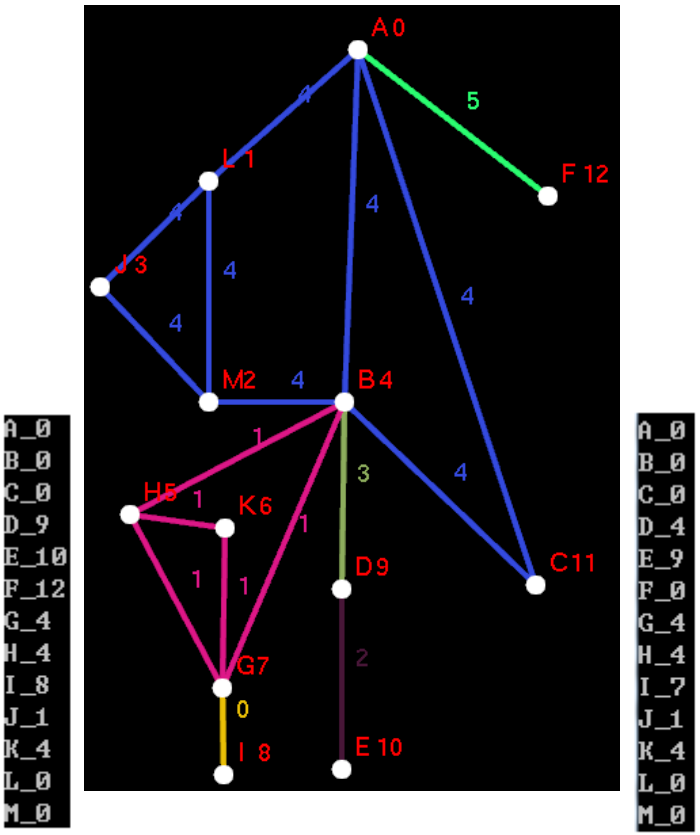
Results


1 0
E 10
5 4
T 9
2 1
4 3
8 7
7 6
6 5
9 8
3 2


A 0
L 1
F 12
J 3
M2
B 4
H5
K6
D9
C11
G7
I 8
E 10


A 0
B 1
H7
C2
G6
M12
I 8
N13
P 15
D3
J 9
K10
L 11
E 4
F 5
O14

计算high值可以摒除指向parent的回边也可以不摒除,对关节点的寻找没有影响,也不影响分支数量的结果. 但是某些点的high值会在摒除时变大.(**结果都是正确的!**)



在计算high值时摒除回边　　在计算high值时不摒除回边

但是在计算边划分时必须摒除指向parent的回边.(**否则结果是错的!**)

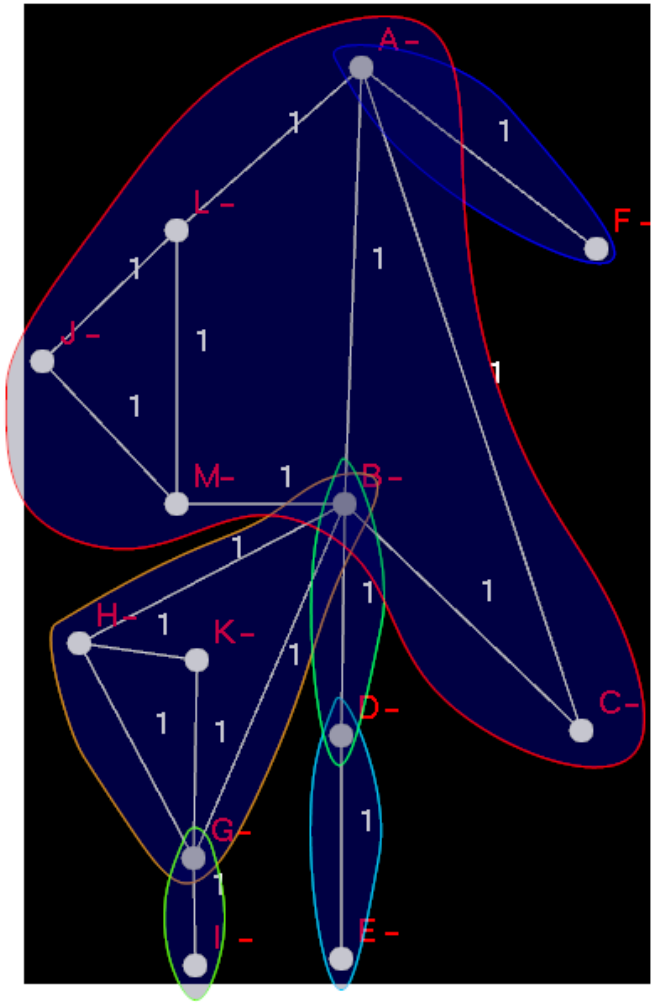
Output bi-components

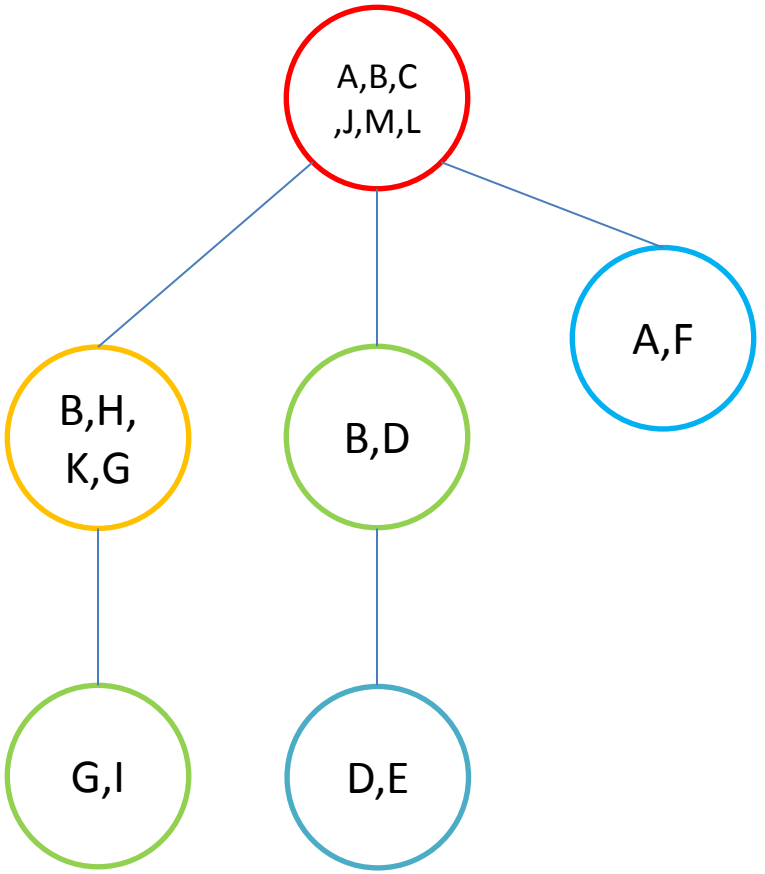
- Use a stack to record only the top-down tree edges(from a parent to a child) and back edges(from a vertex to an ancestor(not the direct parent!! Since otherwise there will be overlap with the top-down tree edges) .

[make sure to push each actual graph edge into the stack for once and only once]

Biconnected-Tree

- Start with a certain components, forming a tree in BFS fashion.
- Components that shares one articulated vertex gives an edge.

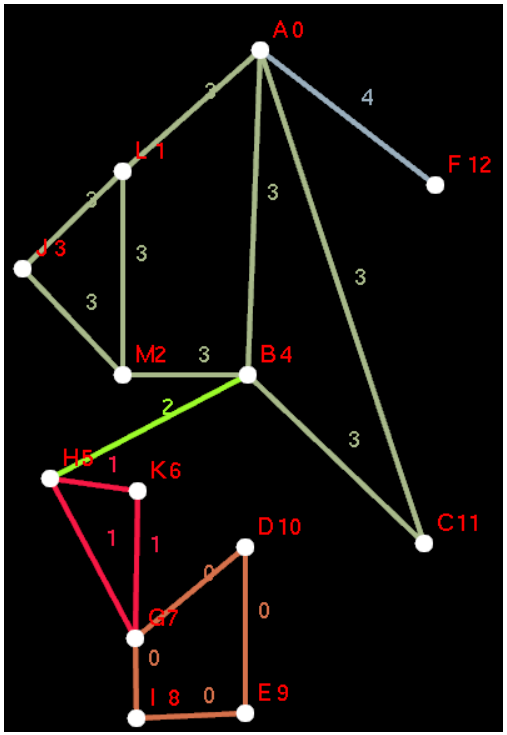


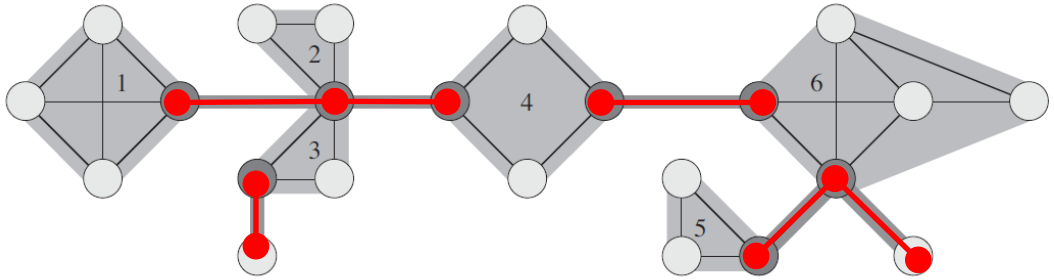


桥

- 桥: 一种边, 删除该边, G的剩余部分不连通.
- 桥的两个端点都是关节点

例如下图中的分支2,





强连通分支

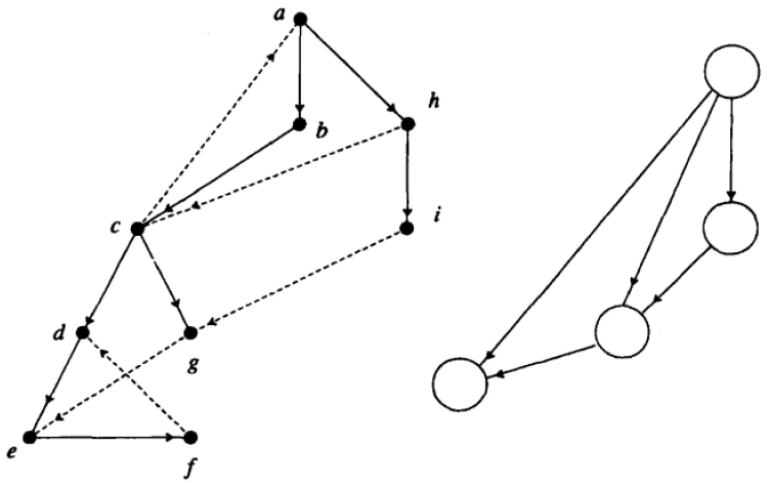
- 针对有向图
- 是一个最大的顶点子集,其导出子图是强连通的.即不存在一个更大的包含它在内的点的子集，其导出子图也是强连通的.

强连通分支划分,唯一存在

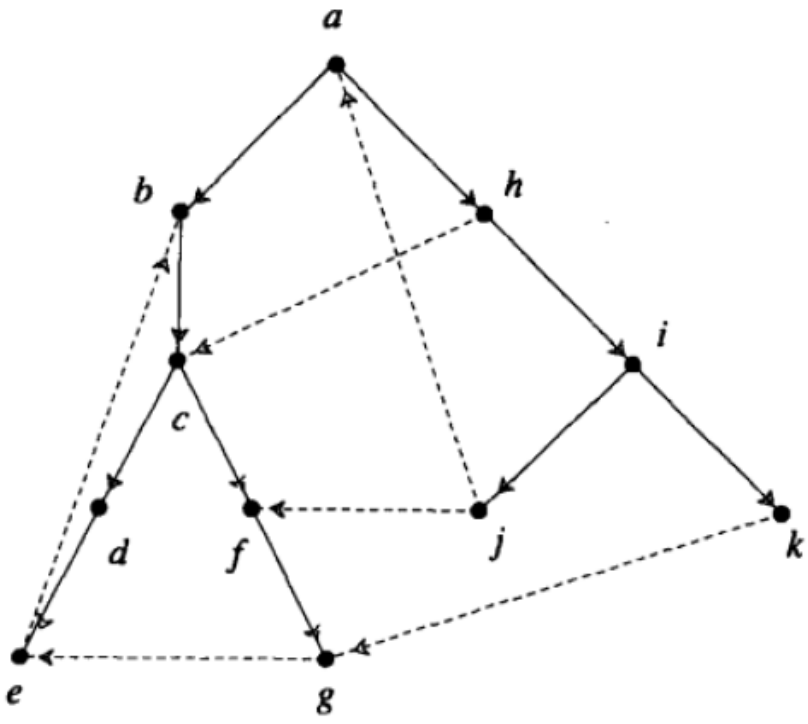
- 引理7.11:两个顶点属于同一个强连通分支，等价于存在一个包含他们的回路.
- 引理7.12:每个顶点仅属于一个强连通分支.

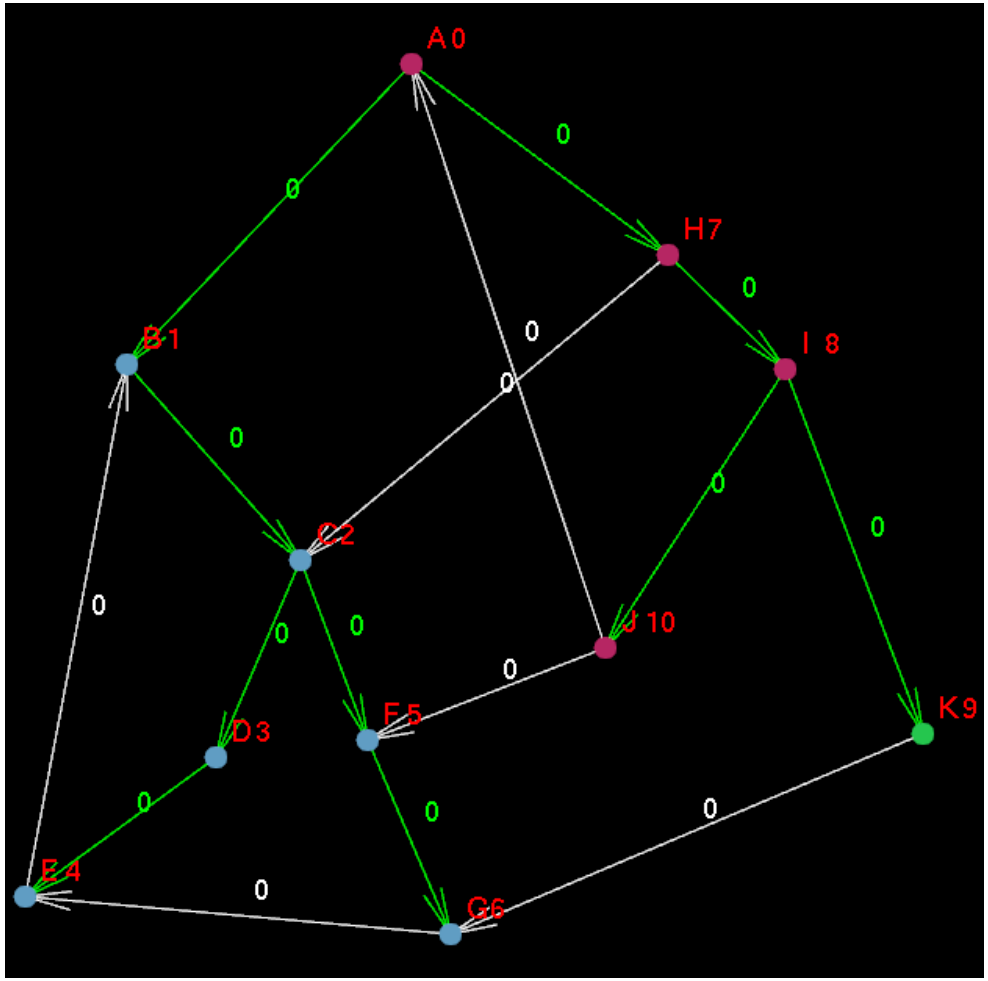
强连通分支的凝聚图
condesation graph

- 一个强连通分支对应于一个点
- 如果分支A上有一个点指向分支B上的一个点, 则在凝聚图上存在A到B一条边.
- 凝聚图是非循环的. 因为如果循环，则可以合成出更大的分支.



result





Output sc-components

- Use a stack to record only vertices in DFS visited order.

[Easier than output biconnected-components]