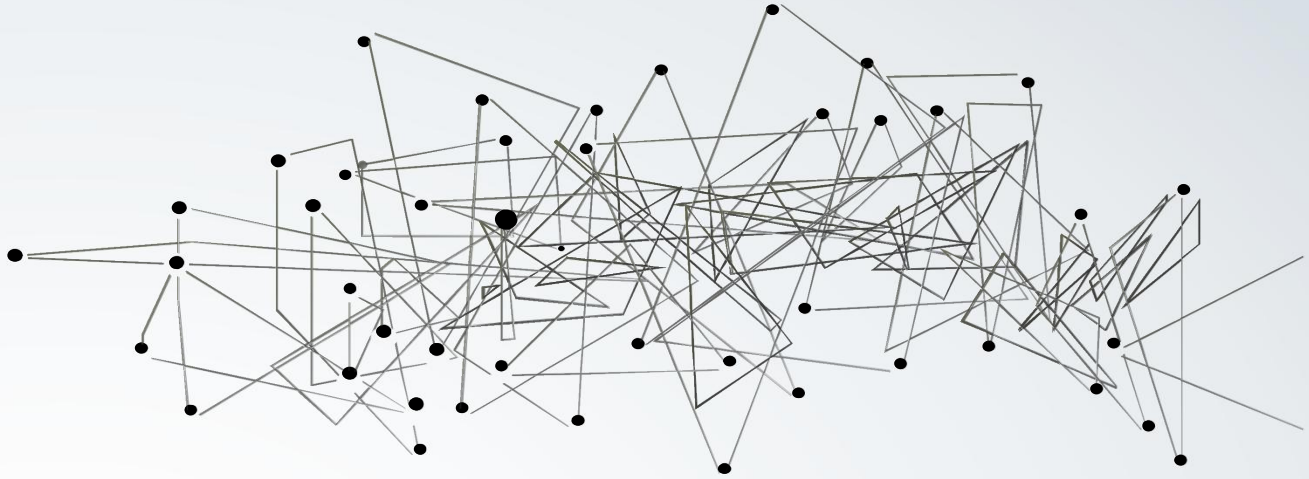
图的应用

3-4 open topic

朱宇博
计算机科学与技术系
14th Oct 2020

CONTENTS



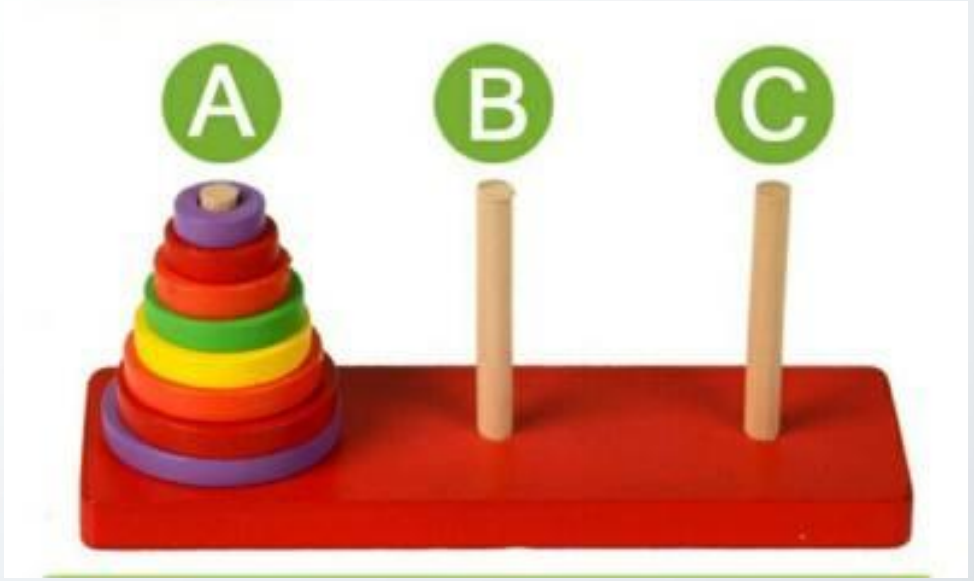
1

First

汉诺塔的图论建模

汉诺塔问题回顾

有三根杆(编号A、B、C)，在A杆自下而上、由大到小按顺序放置64个金盘(如下图)。游戏的目标：把A杆上的金盘全部移到C杆上，并仍保持原有顺序叠好。操作规则：每次只能移动一个盘子，并且在移动过程中三根杆上都始终保持大盘在下，小盘在上，操作过程中盘子可以置于A、B、C任一杆上。



汉诺塔问题建模

如何来刻画当前的状态？

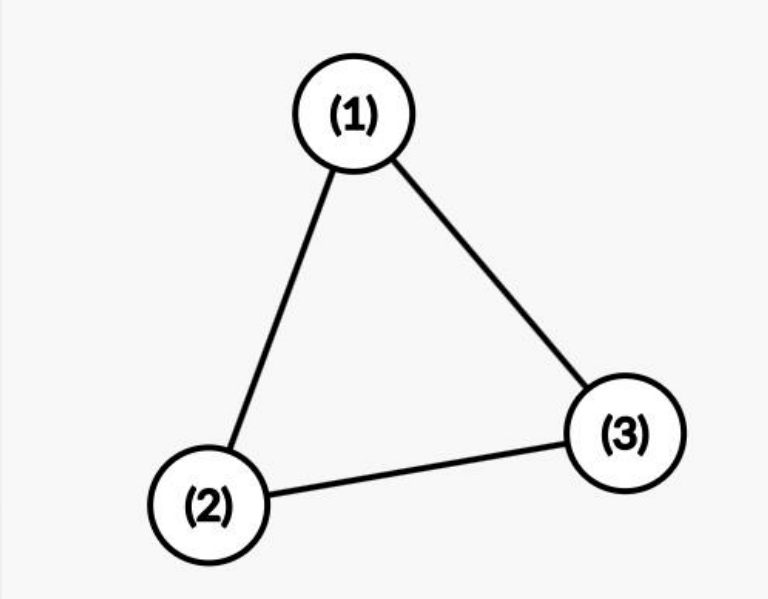
需要关注的信息：每一个盘子当前在哪根柱子

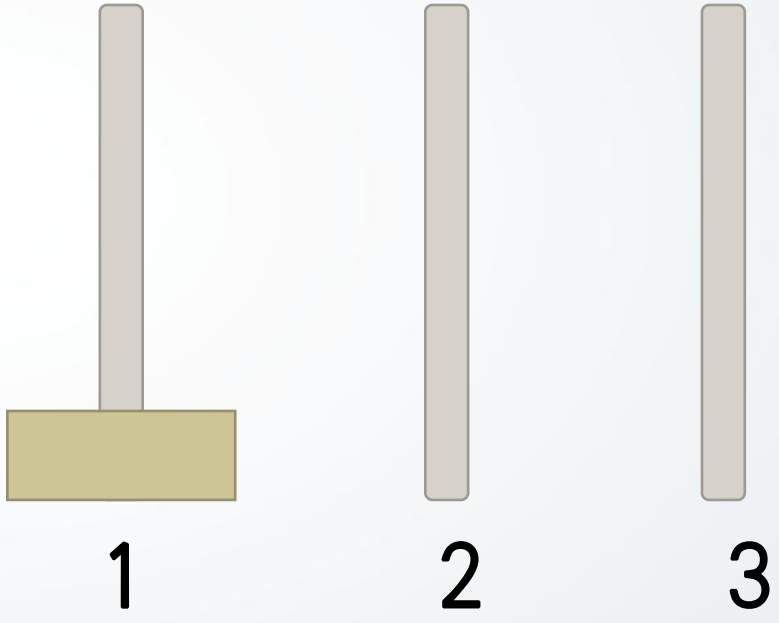
假设有n个盘子

用(1,1,...,1)为初始状态，则最终状态为(3,3,...,3)
其中，第i位表示第i小的盘子当前的位置

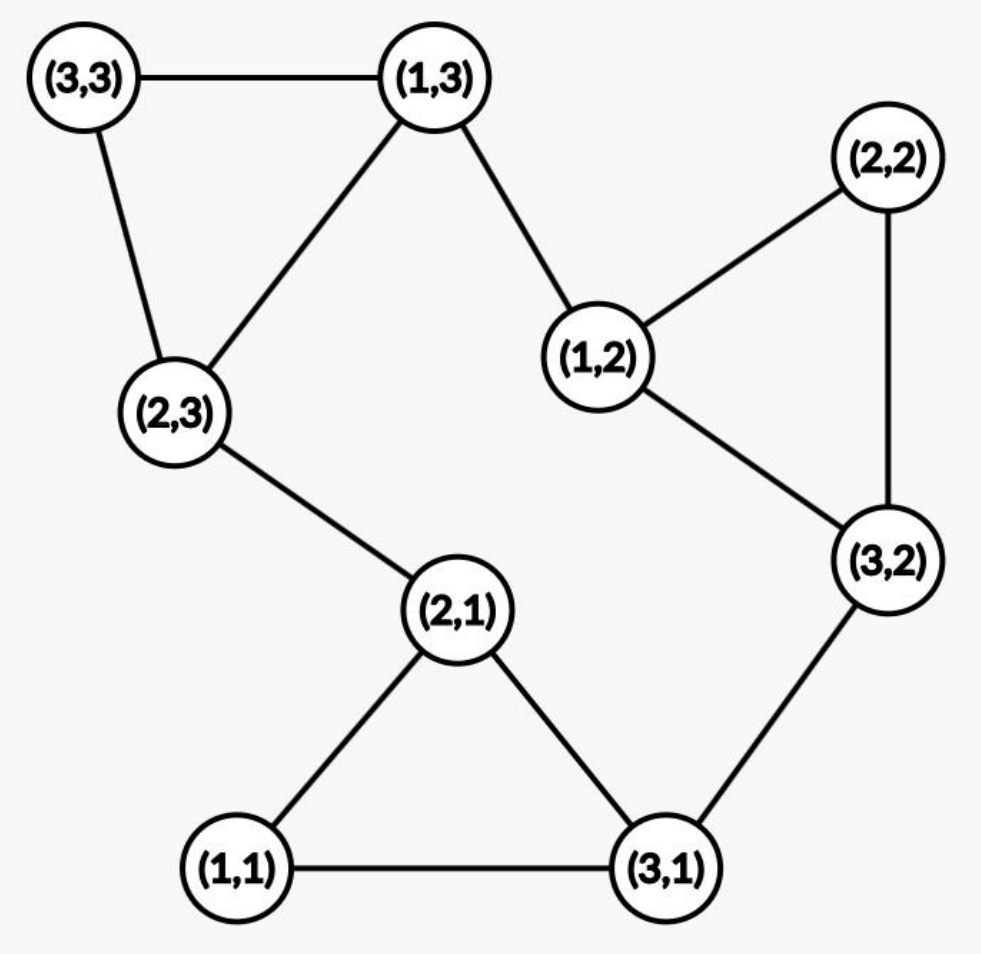
移动规则：每次仅移动一个盘子，且移动的盘子必为当前柱子上最小的盘子

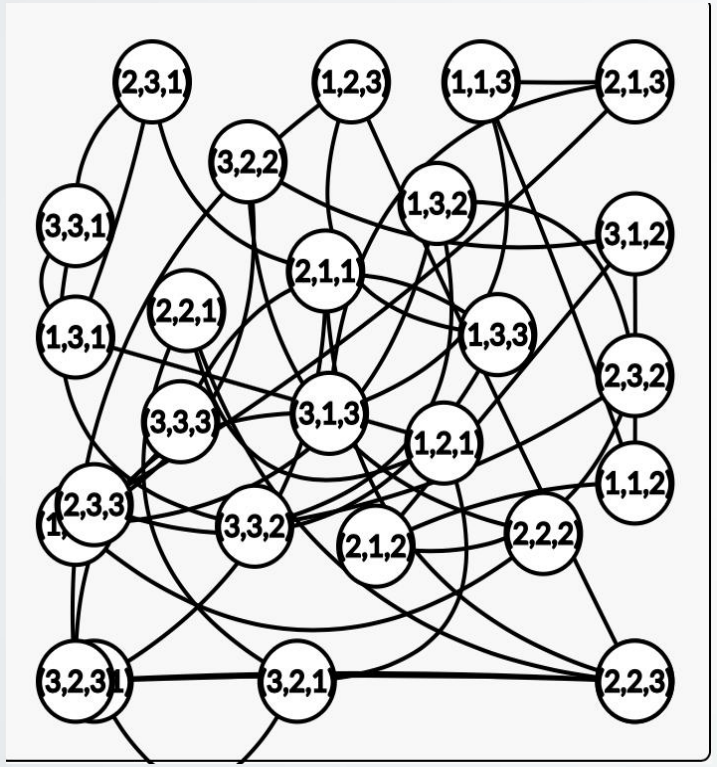
1个圆盘



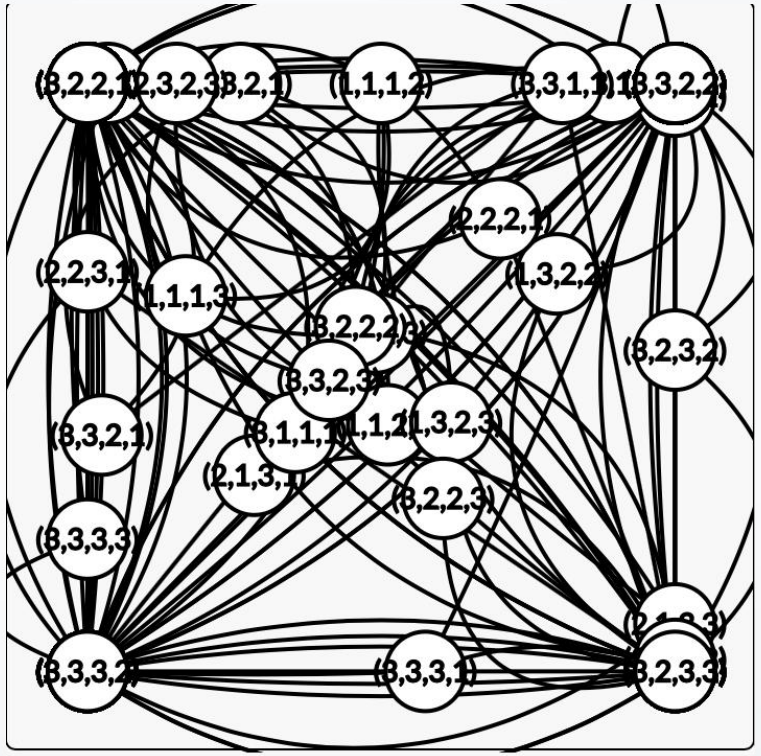


2个圆盘

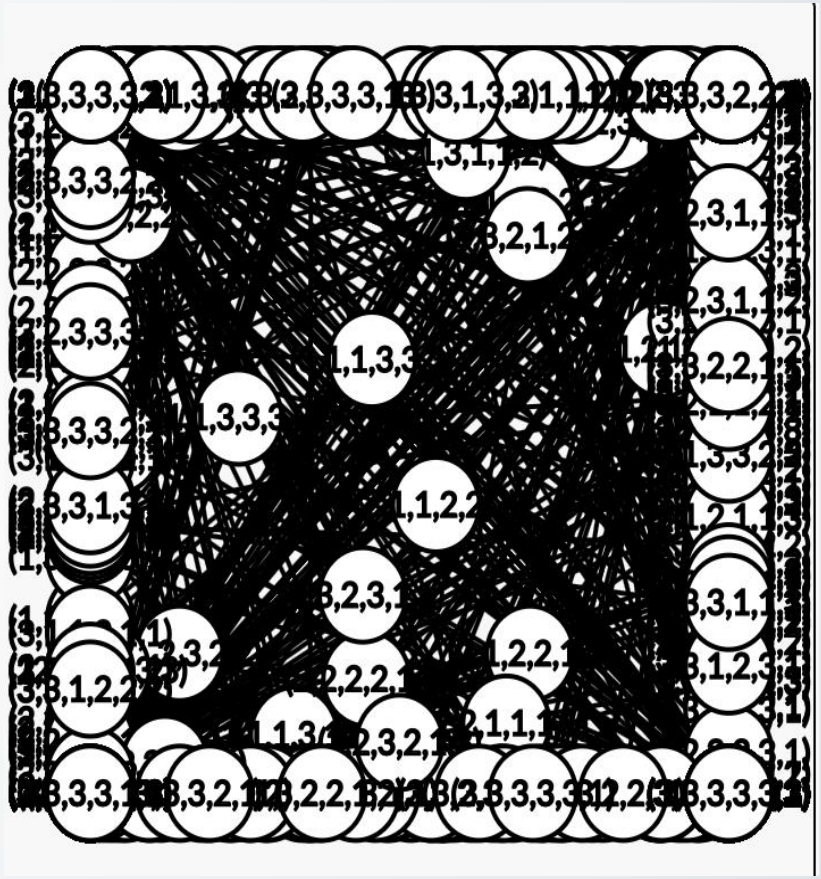



2,3,1
1,2,3
1,1,3
2,1,3
3,2,2
1,3,2
3,3,1
3,1,2
2,1,1
2,2,1
1,3,1
1,3,3
2,3,2
3,3,3
3,1,3
1,2,1
1,1,2
2,3,3
3,3,2
2,1,2
2,2,2
3,2,3
3,2,1
2,2,3

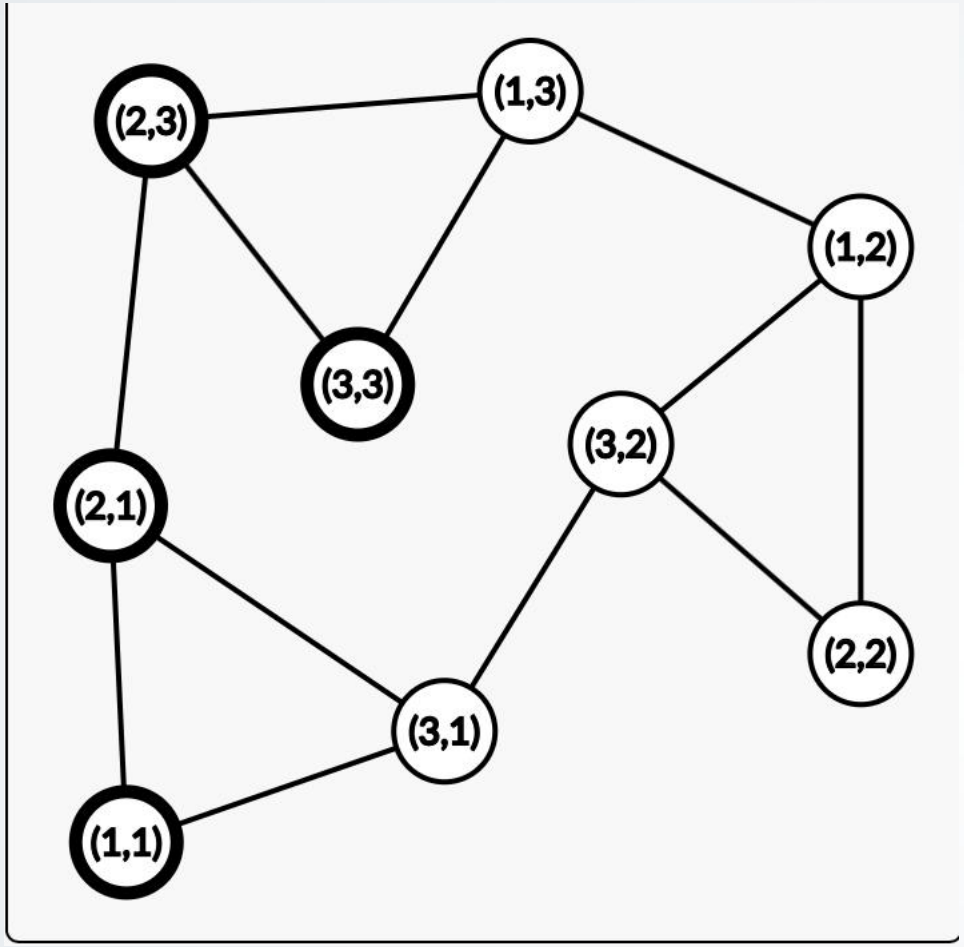
n=3

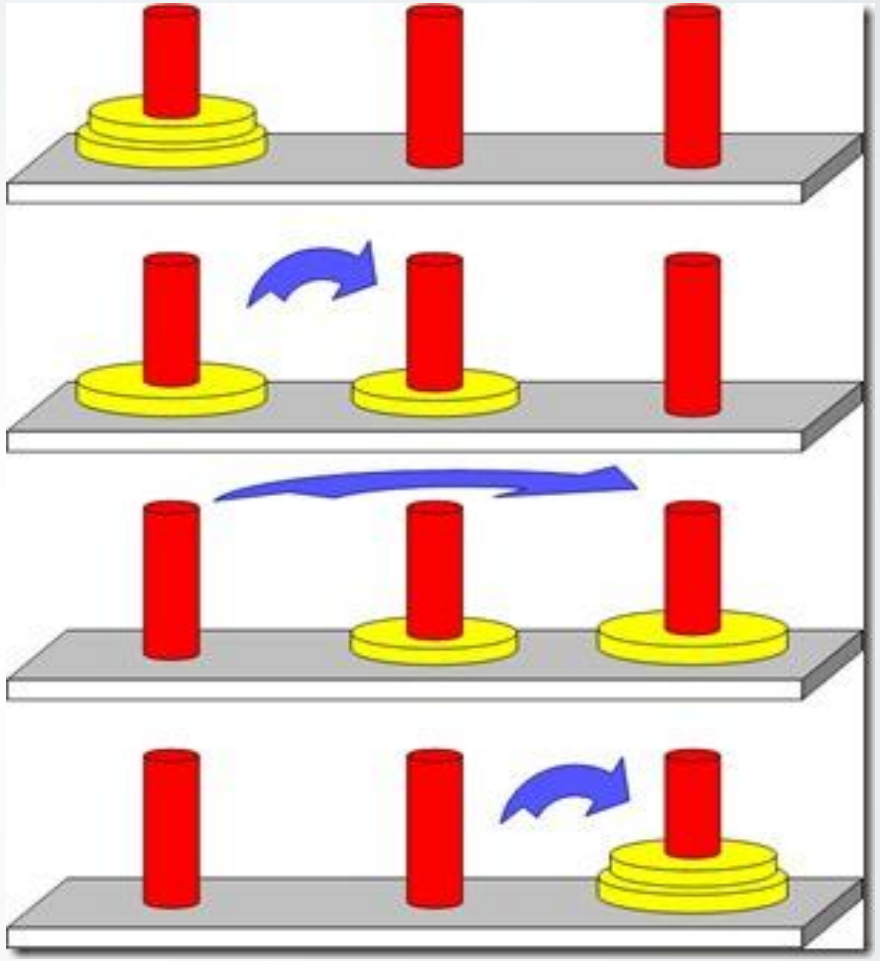

(3,2,2,1)
(2,3,2,3)
(3,2,1)
(1,1,1,2)
(3,3,1,1)
(3,3,2,2)
(2,2,2,1)
(2,2,3,1)
(1,1,1,3)
(1,3,2,2)
(3,2,2,3)
(3,2,3,2)
(3,3,2,3)
(3,2,3,1)
(3,1,1,1)
(1,1,2,2)
(1,3,2,3)
(2,1,3,1)
(3,2,2,3)
(3,3,3,3)
(3,3,3,2)
(3,3,3,1)
(2,2,3,3)

n=4



n=5





汉诺塔的最优解对应着一条从（1，1）到（3，3）的最短路

something interesting

显然，点的数目为3^n
那么边的数量呢？

3,12,39,120,363,1092 Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

Search: **seq:3,12,39,120,363,1092**

Displaying 1-2 of 2 results found. page 1

Sort: relevance | references | number | modified | created Format: long | short | data

A029858 a(n) = (3^n - 3)/2. +30 27

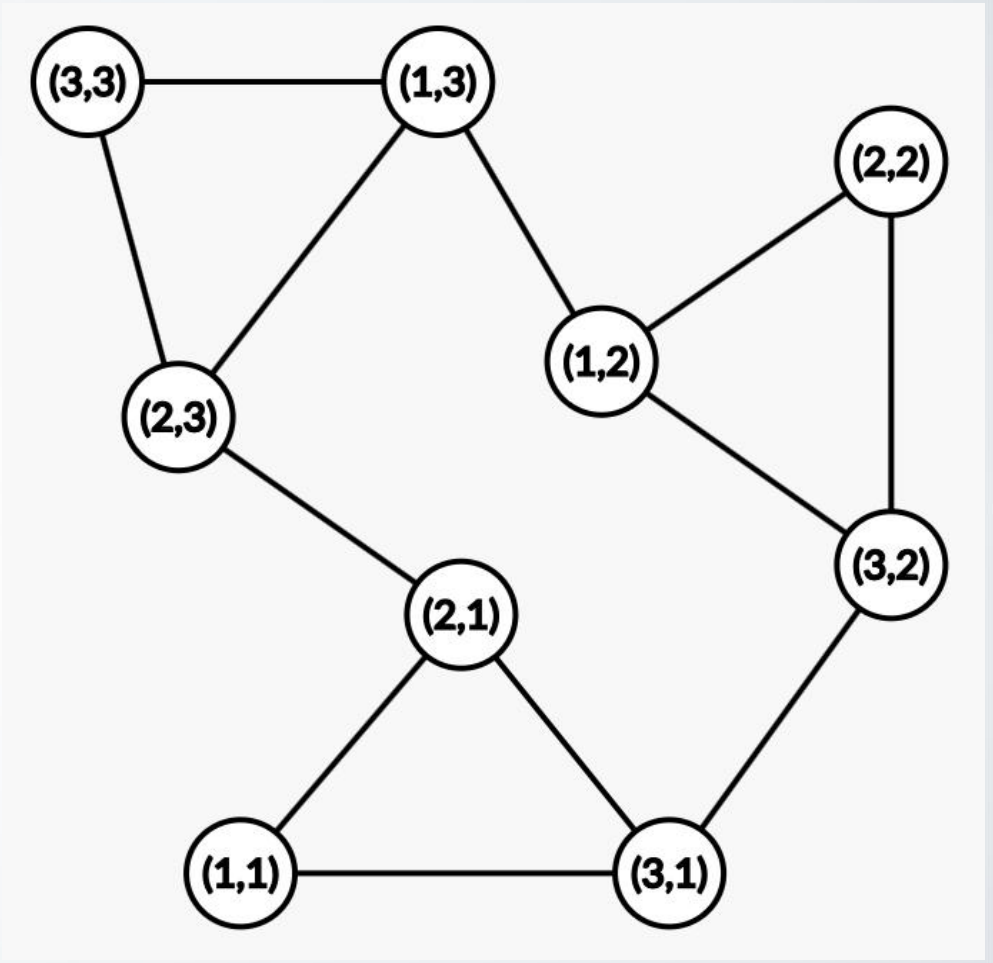
0, **3, 12, 39, 120, 363, 1092**, 3279, 9840, 29523, 88572, 265719, 797160, 2391483, 7174452, 21523359, 64570080, 193710243, 581130732, 1743392199, 5230176600, 15690529803, 47071589412, 141214768239 (list; graph; refs; listen; history; text; internal format)

$$\frac{3^{n+1}-3}{2} = 3\frac{3^n-1}{2} = \frac{3}{2}3^n - \frac{3}{2}$$

$$\frac{3^{n+1}-3}{2}=3\frac{3^n-1}{2}=\frac{3}{2}3^n-\frac{3}{2}$$

平均每个点连出去3/2条边？
再减去每个三元环中，都有1个点少连1条边？

平均每个点连出去3/2条边？
再减去(1,1,...,1).(2,2,...,2)...(n,n...,n)



1．对于每一个状态，我们不失一般性地假设其最小的盘子在A柱上。

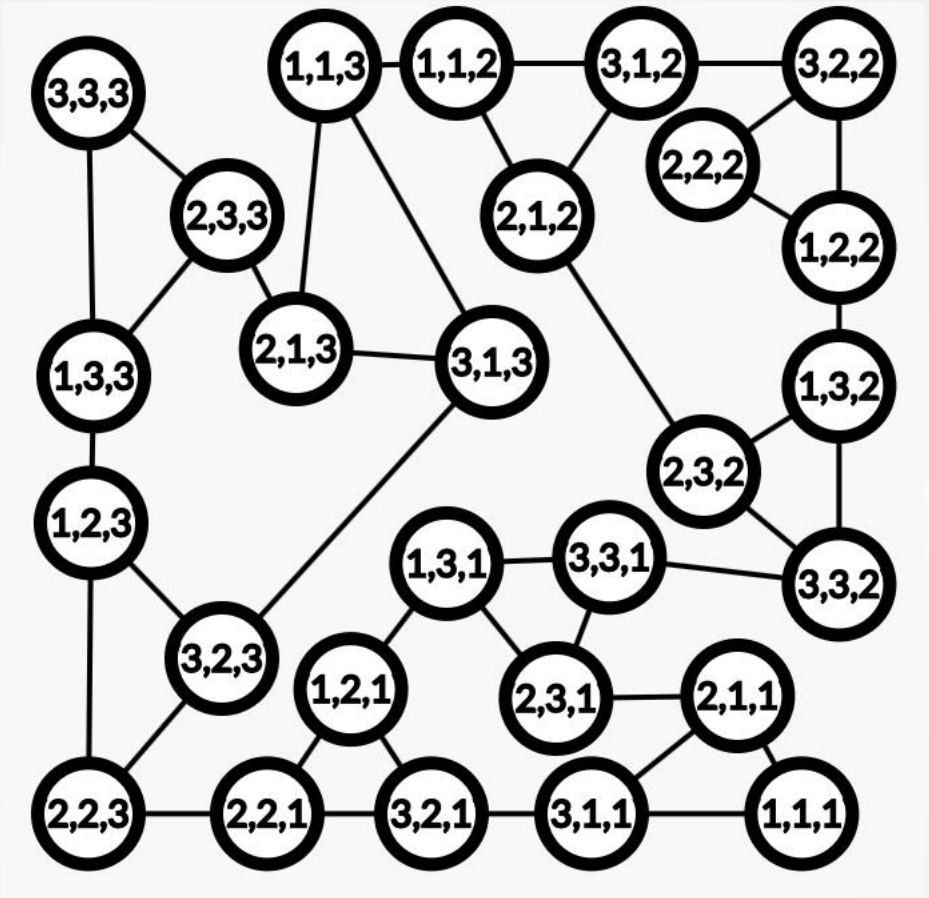
则该状态、将最小的盘子移到B柱、将最小的盘子移到C柱三个状态间构成三元环

2．对于每一个状态

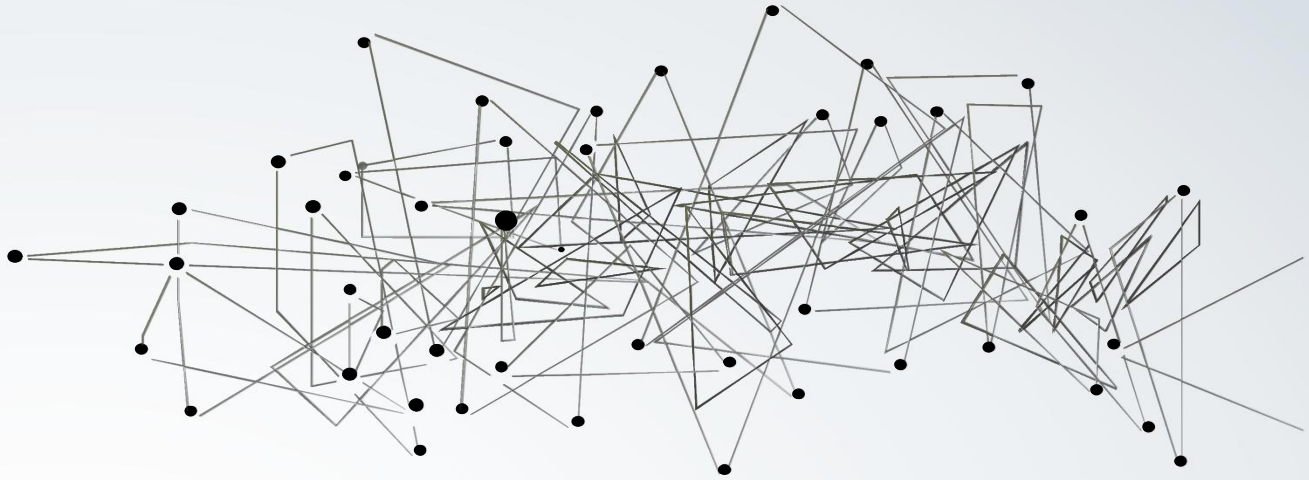
若其盘子不全在一个柱子上，我们不失一般性地假设其最小的盘子在A柱上，且除去A柱外，最小的盘子在B柱上。则该状态到将B柱最小盘移动向C柱的状态转移，构成三元环的导出边。

若其盘子全在一个柱子上，则该状态只会在1中构成三元环，没有导出边

综上，n个盘子的是有3个偶点，n−3个奇点，n/3个三元环的连通图


3,3,3
1,1,3
1,1,2
3,1,2
3,2,2
2,2,2
2,3,3
2,1,2
1,2,2
2,1,3
3,1,3
1,3,3
1,3,2
2,3,2
1,2,3
1,3,1
3,3,1
3,3,2
3,2,3
1,2,1
2,3,1
2,1,1
2,2,3
2,2,1
3,2,1
3,1,1
1,1,1

整理后的n=3的图



2

Second

PageRank算法

人工分类目录 --> 文本分析时代 --> 链接分析阶段 --> 用户行为分析阶段

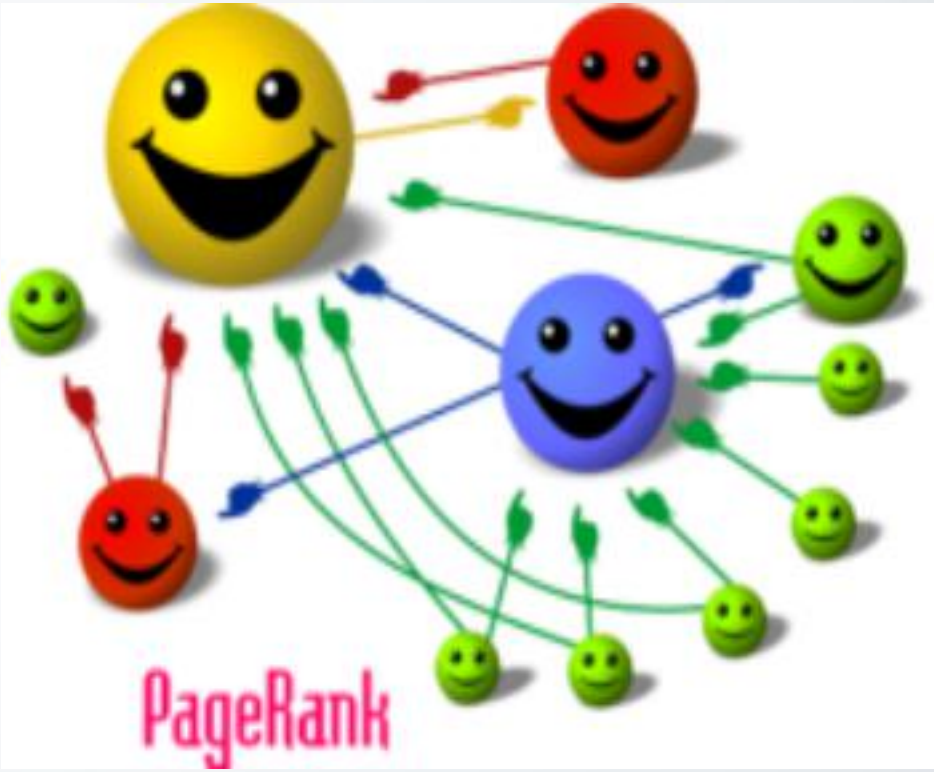
PageRank算法

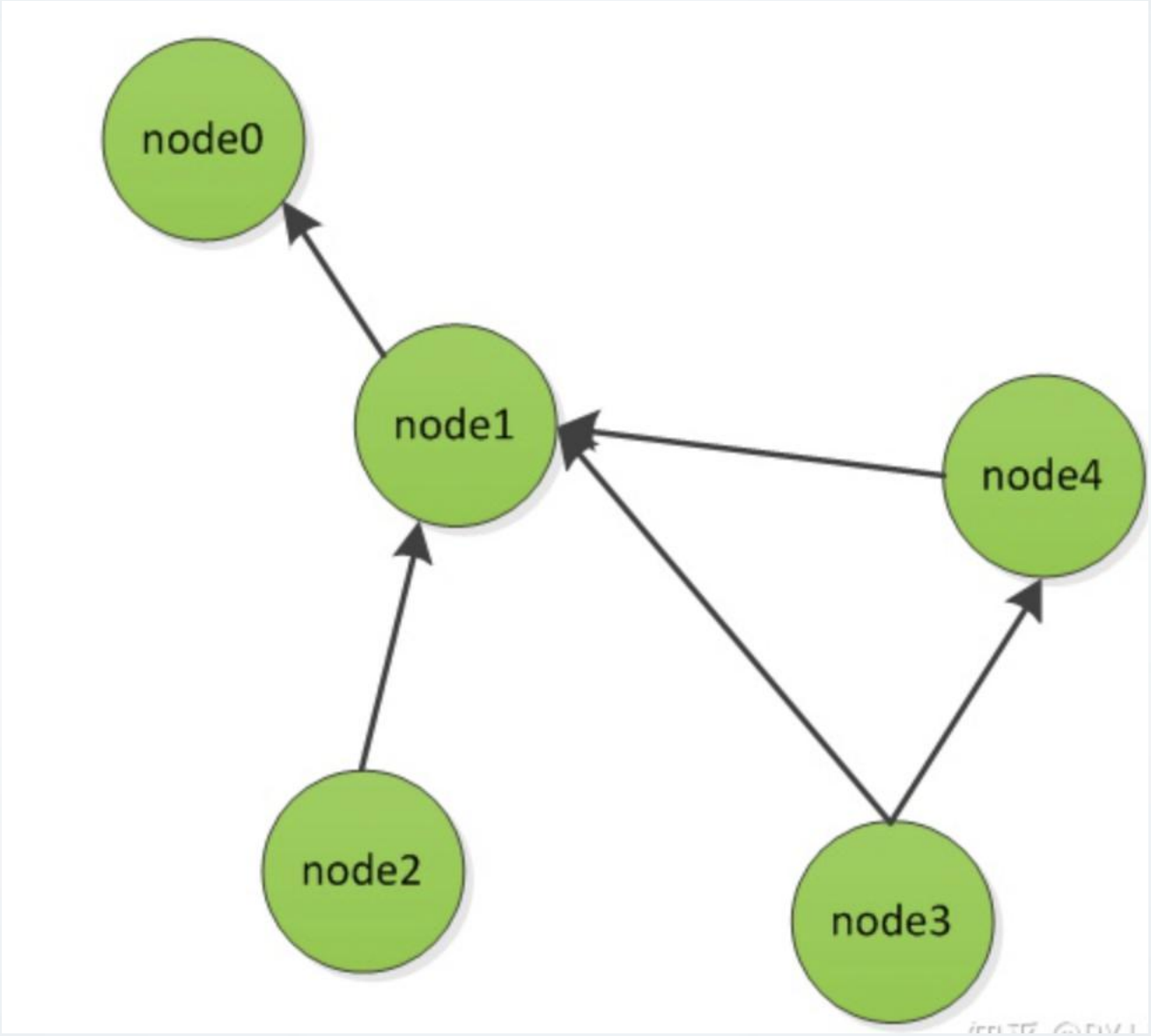
PageRank（PR）是Google搜索用来在搜索引擎结果中对网页进行排名的算法，它是一种衡量网站页面重要性的方法。

PageRank算法的核心思想

1， 如果一个网页被很多其他网页链接到的多，则说明这个网页比较重要，也就是PageRank值会相对较高

2， 如果一个PageRank值很高的网页链接到其他的网页，那么被链接到的网页的PageRank值会相应地因此而提高

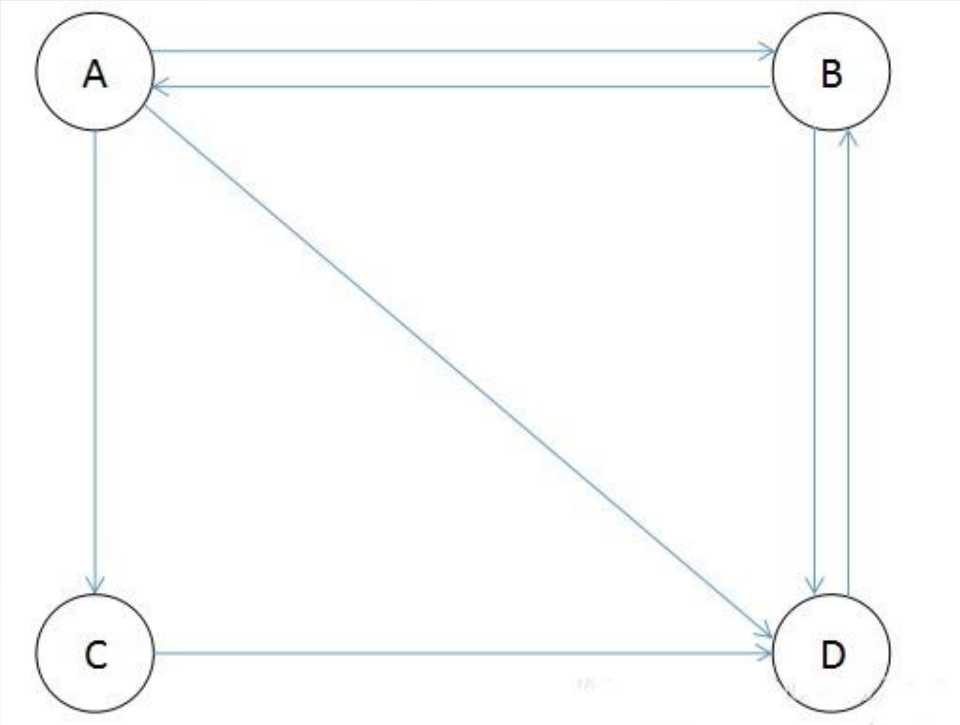



node0
node1
node4
node2
node3

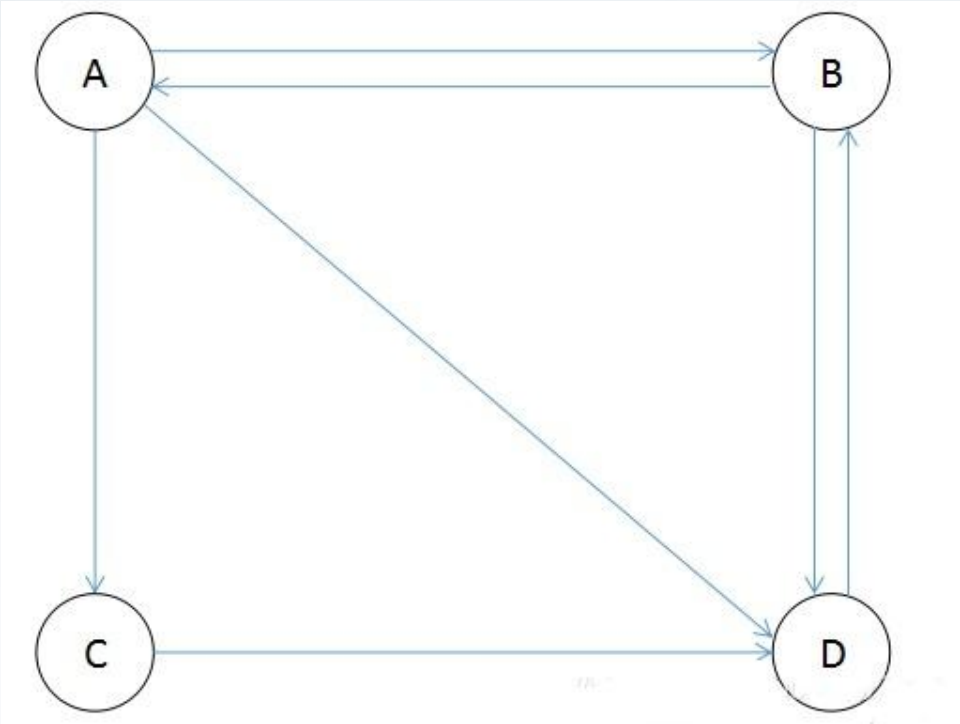
PageRank算法

1，给每个网页一个PR值（下面用PR值指代PageRank值）

2，通过（投票）算法不断迭代，直至达到平稳分布为止。



在一开始时，有PR(A) = PR(B) = PR(C)= PR(D) = 0.25



根据图中的导出情况，有以下关系：

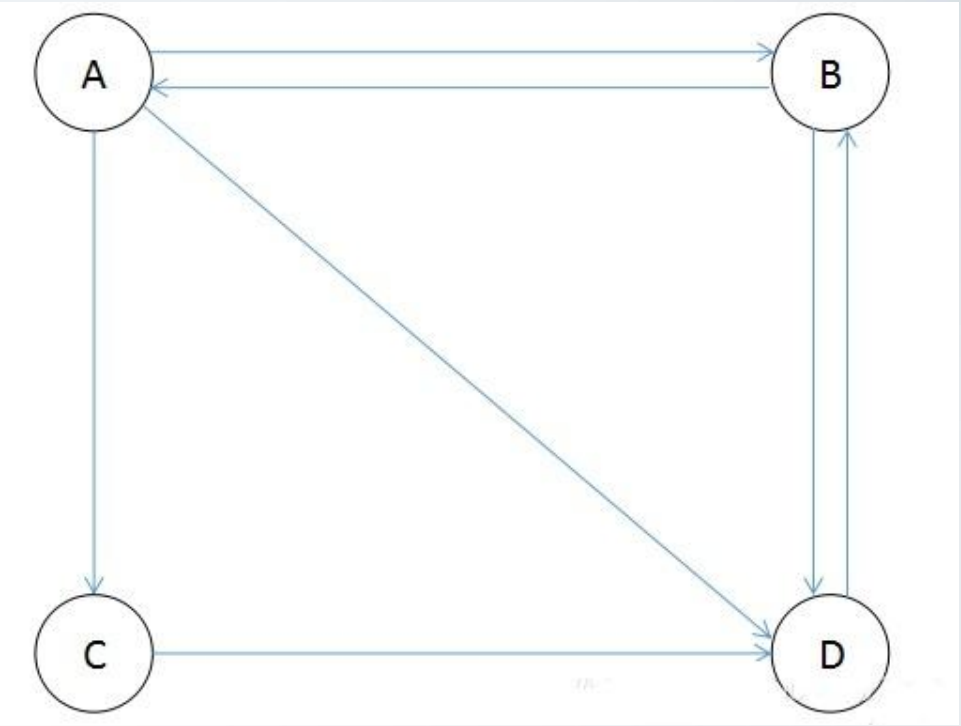
PR(A)=PR(B)/2

PR(B)=PR(A)/3+PR(D)

PR(C)=PR(A)/3

PR(D)=PR(A)/3+PR(B)/2+PR(C)

PR(A)=PR(B)/2
PR(B)=PR(A)/3+PR(D)
PR(C)=PR(A)/3
PR(D)=PR(A)/3+PR(B)/2+PR(C)

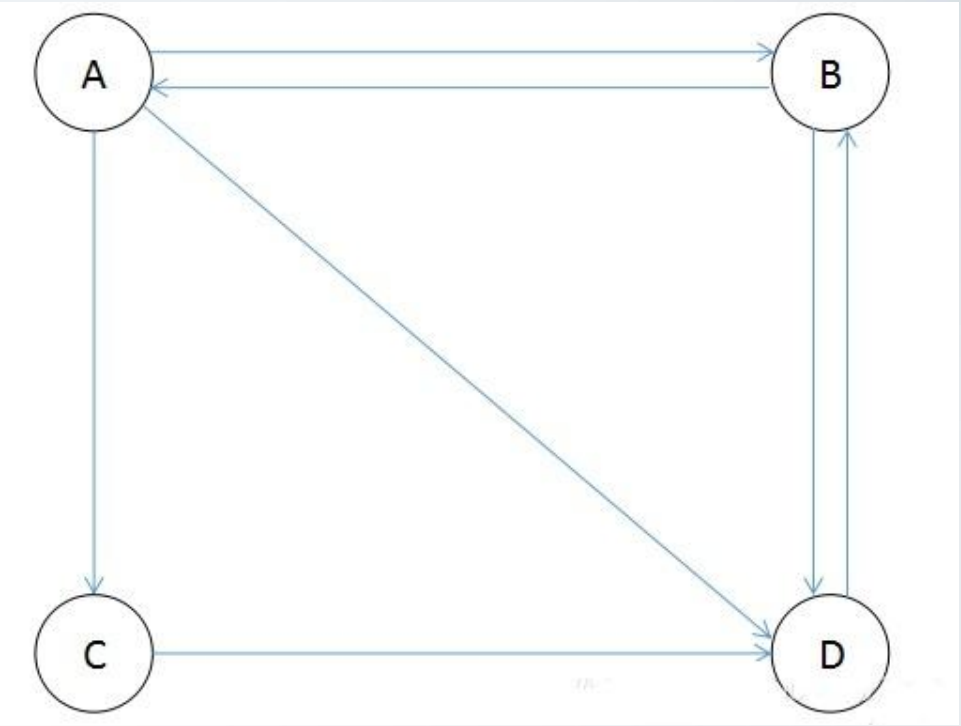


在经过一轮迭代后，

0：PR(A) = 0.25 PR(B) = 0.25 PR(C) = 0.25 PR(D) = 0.25

1：PR(A) = 0.125 PR(B) = 0.333 PR(C) = 0.083 PR(D) = 0.458

PR(A)=PR(B)/2
PR(B)=PR(A)/3+PR(D)
PR(C)=PR(A)/3
PR(D)=PR(A)/3+PR(B)/2+PR(C)



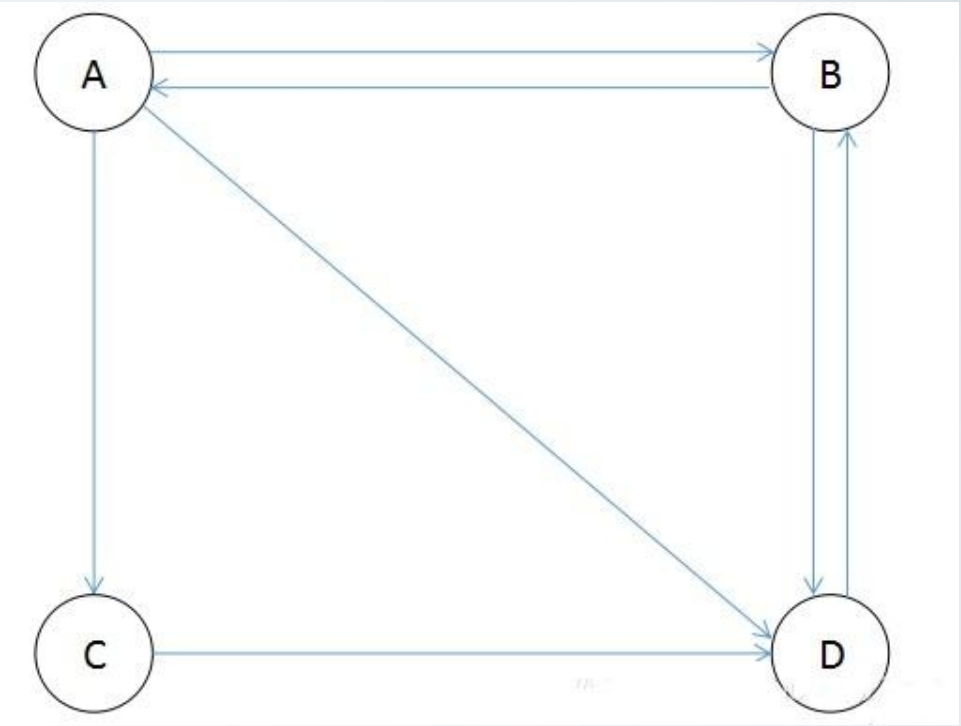
在经过两轮迭代后，

0：PR(A) = 0.25 PR(B) = 0.25 PR(C) = 0.25 PR(D) = 0.25

1：PR(A) = 0.125 PR(B) = 0.333 PR(C) = 0.083 PR(D) = 0.458

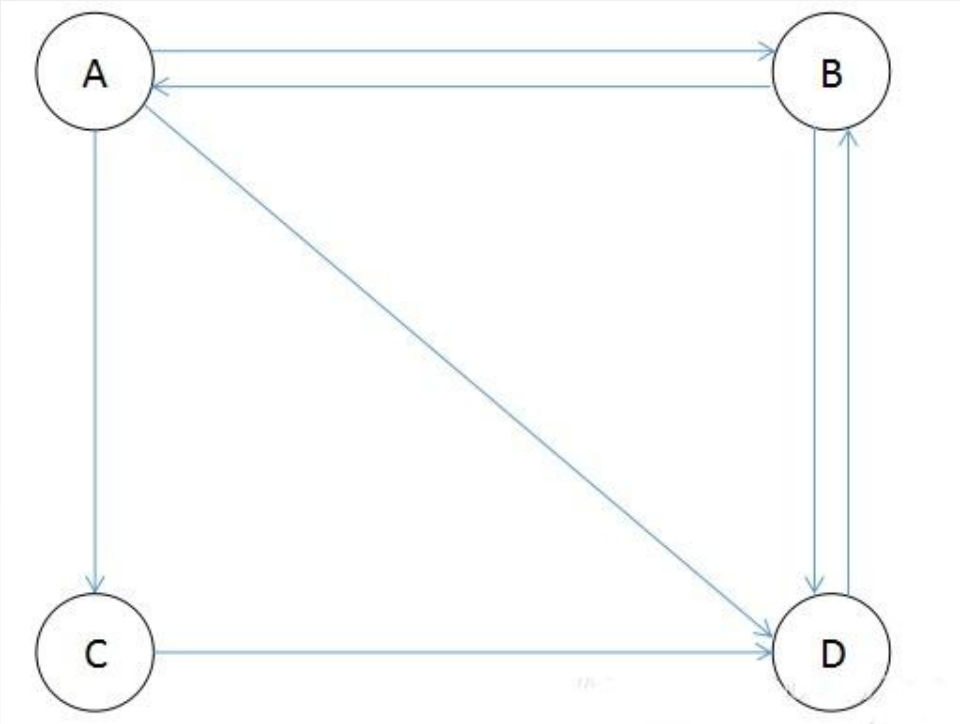
2：PR(A) = 0.166 PR(B) = 0.499 PR(C) = 0.041 PR(D) = 0.291

PR(A)=PR(B)/2
PR(B)=PR(A)/3+PR(D)
PR(C)=PR(A)/3
PR(D)=PR(A)/3+PR(B)/2+PR(C)



在经过n轮迭代后，

0：PR(A) = 0.25 PR(B) = 0.25 PR(C) = 0.25 PR(D) = 0.25
1：PR(A) = 0.125 PR(B) = 0.333 PR(C) = 0.083 PR(D) = 0.458
2：PR(A) = 0.166 PR(B) = 0.499 PR(C) = 0.041 PR(D) = 0.291
...
n：PR(A) = 0.199 PR(B) = 0.399 PR(C) = 0.066 PR(D) = 0.333



根据图中的导出情况，有以下关系：

PR(A)=PR(B)/2

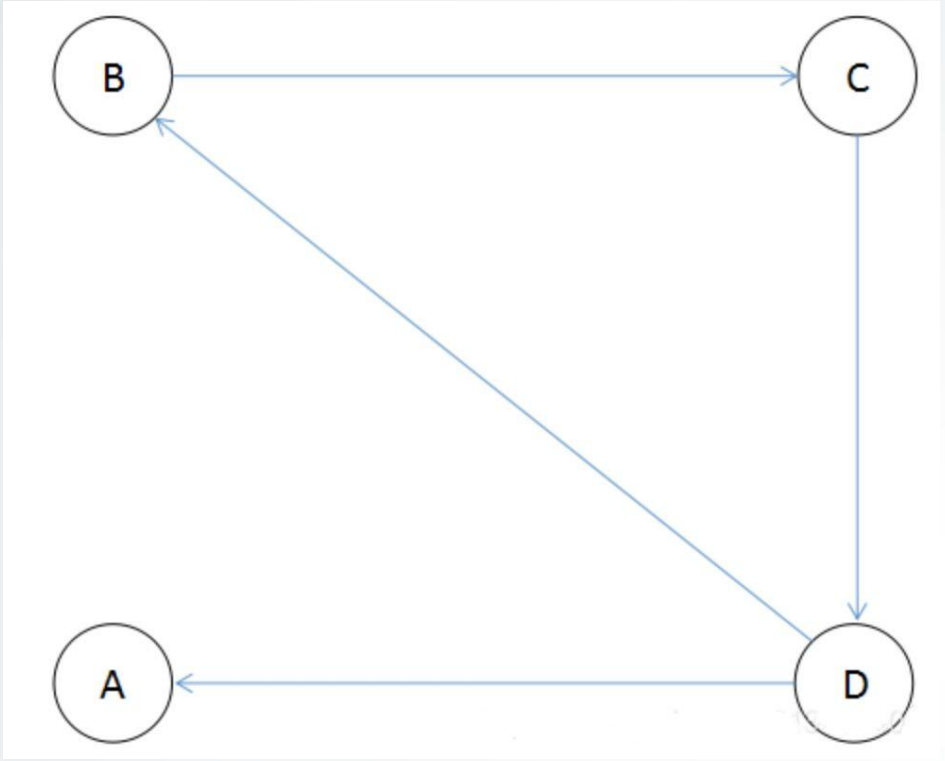
PR(B)=PR(A)/3+PR(D)

PR(C)=PR(A)/3

PR(D)=PR(A)/3+PR(B)/2+PR(C)

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

其中，Bu是连向u的点的集合，L(v)代表结点v的出度



	PR(A)	PR(B)	PR(C)	PR(D)
初始值	0.25	0.25	0.25	0.25
一次迭代	0.125	0.125	0.25	0.25
二次迭代	0.125	0.125	0.125	0.25
三次迭代	0.125	0.125	0.125	0.125
……	……	……	……	……
n次迭代	0	0	0	0

若存在结点只有入度，没有出度，该如何处理？
若又存在点只有出度，没有入度，又该如何处理？

PageRank算法的随机浏览模型

假设了这样一个场景：用户并不都是按照跳转链接的方式来上网，还有一种可能是不论当前处于哪个页面，都有概率访问到其他任意的页面，比如说用户就是要直接输入网址访问其他页面，虽然这个概率比较小。

增加阻尼系数d

$$PR(u) = \frac{1-d}{N} + d \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

$$PR(u) = \frac{1-d}{N} + d\sum_{v \in B_u} \frac{PR(v)}{L(v)} = \frac{1-d}{N} + d\sum_{v \in B_u} PR(v)\frac{1}{L(v)}$$

定义函数 $l(u,v) = \begin{cases} \frac{1}{L(v)} & \text{e(v,u)=1} \\ 0 & \text{e(v,u)=0} \end{cases}$

$$PR(u) = \frac{1-d}{N} + d\sum_{v=1}^{n} PR(v)l(u,v)$$

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

$$PR(u) = \frac{1-d}{N} + d\sum_{v=1}^{n} PR(v)l(u,v)$$

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

$$\mathbf{R}(t+1) = d\mathbf{M}\mathbf{R}(t) + \frac{1-d}{N}\mathbf{E}$$

用矩阵的形式表示迭代，则有

$$\mathbf{R}(t+1) = d\mathbf{M}\mathbf{R}(t) + \frac{1-d}{N}\mathbf{E}$$

其中矩阵M为

$$M_{ij} = \begin{cases} \frac{1}{L(p_j)} & \mathrm{e(j,i)=1} \\ 0 & \mathrm{e(j,i)=0} \end{cases}$$

其中矩阵E为 $[1, 1, ..., 1]^T$

$$\mathbf{R}(t+1) = d\mathbf{M}\mathbf{R}(t) + \frac{1-d}{N}\mathbf{E}$$

$$= d\mathbf{M}\mathbf{R}(t) + \frac{1-d}{N}\mathbf{E}\mathbf{E}^T\mathbf{R}(t)$$

$$= (d\mathbf{M} + \frac{1-d}{N}\mathbf{E}\mathbf{E}^T)\mathbf{R}(t)$$

令 $\mathbf{A} = d\mathbf{M} + \frac{1-d}{N}\mathbf{E}\mathbf{E}^T$

则有 $\mathbf{R}(t+1) = \mathbf{A}\mathbf{R}(t)$

$$\mathbf{R}(t+1) = \mathbf{A}\mathbf{R}(t)$$

计算PR值的过程变成了Markov过程

如果一个Markov过程收敛，则状态转移矩阵A需要满足：

1，A为随机矩阵。
2，A是不可约的。
3，A是非周期的。

$$\mathbf{R}(t+1) = \mathbf{A}\mathbf{R}(t)$$

当 $t \to \infty$

$$\mathbf{R} = d\mathbf{M}\mathbf{R} + \frac{1-d}{N}\mathbf{E}$$

$$\mathbf{R} = (\mathbf{I} - d\mathbf{M})^{-1}\frac{1-d}{N}\mathbf{E}$$

至此，我们便求出了无限次迭代后PR值

PageRank算法的缺陷

1. 广告和功能链接无法过滤，即前者链接到一个广告页面，后者一般链接到社交网站

2. 新网页PR值都不高，新网页导入链接相比较少，即便网页内容的质量很高，但要变成一个页面高PR值就很漫长了

由于PageRank算法有明显的缺点，经发展随即产生了TrustRank算法

Thanks

1，感谢陶老师和马老师的指导
2，感谢陈弘毅同学的帮助

THANKS for Listening