链接分析与网络搜索

提要

- 网页排序（ranking、排名、排位）
  - 搜索服务的基本问题，传统信息检索技术的不足
- 中枢与权威
  - 一篇网页的两面性；有向图的启示
- 中枢值与权威值的计算（HITS算法）
- PageRank（含义）
- PageRank（计算）
- 退化图结构带来的问题
- 随机游走及其与PageRank定义的等价关系

搜索引擎关心的基本问题

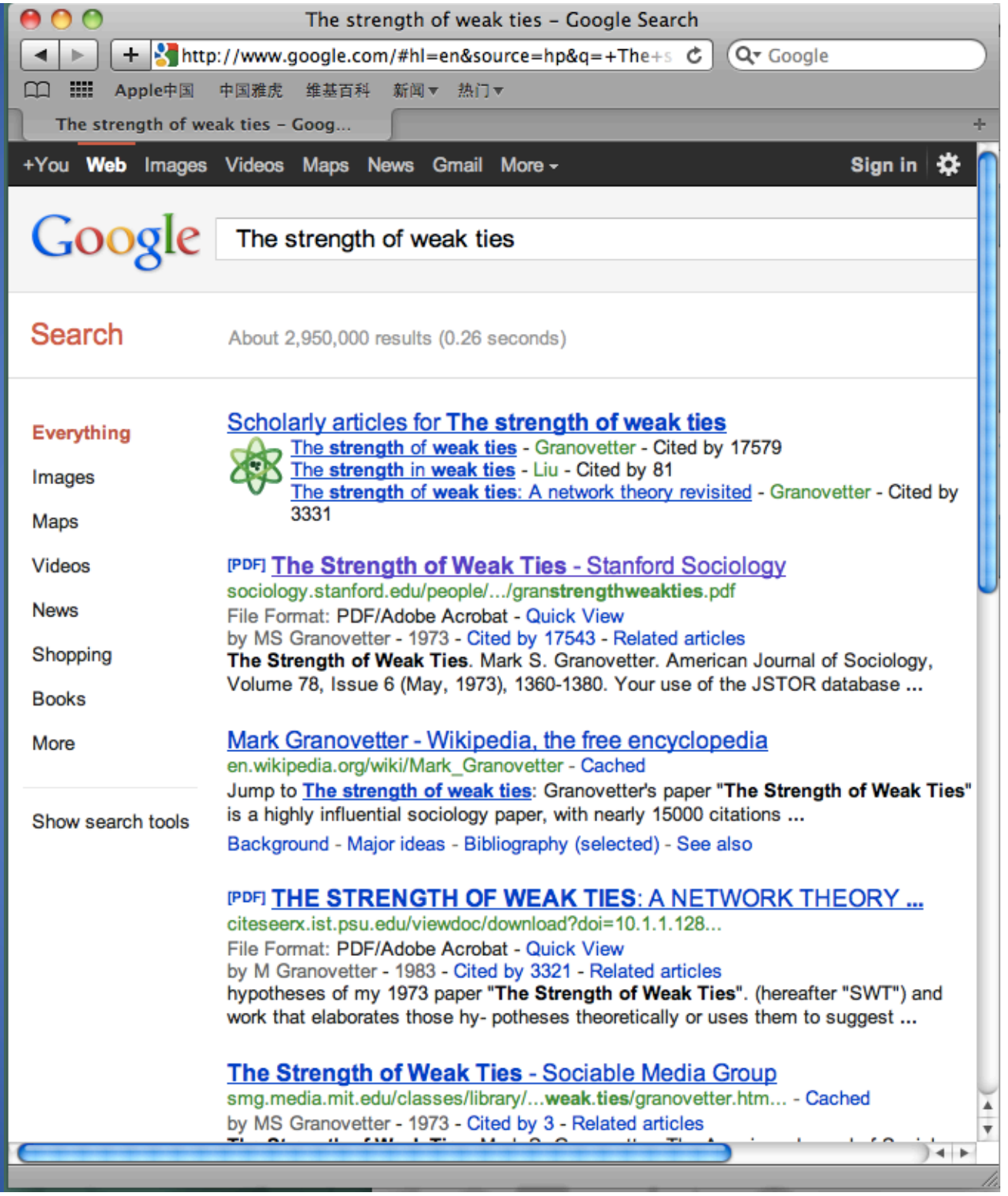
- 计算机显示屏一次只能显示5-6个结果，典型搜索引擎掌握的网页超过10亿
- 对用户提交的一个查询，如何从这种海量网页集合中将最可能满足用户需求的少数几个结果找出来，展现在计算机显示屏上？
  - “最可能满足”的多义性
    - 同一个查询，不同的需求（苹果，病毒等）；
    - 不同的查询，相同的需求（电脑，计算机等）

传统信息检索技术的要点
（information retrieval，IR）

- 基于词语之间的相关性（**relevance**）
  - **similarity(q, d) ≈ Σ score(d, term)**
- 传统应用背景
  - 文档集合：图书，规范的文献
  - 查　询：主题词，关键词
  - 查询意图：获取与查询词有关的书籍和文章
  - 用　户：图书管理人员
- “查询目标包含查询词”是一个合理假设
  - 在形成查询词的时候就有这样的潜意识

现在查找学术文献有类似预期

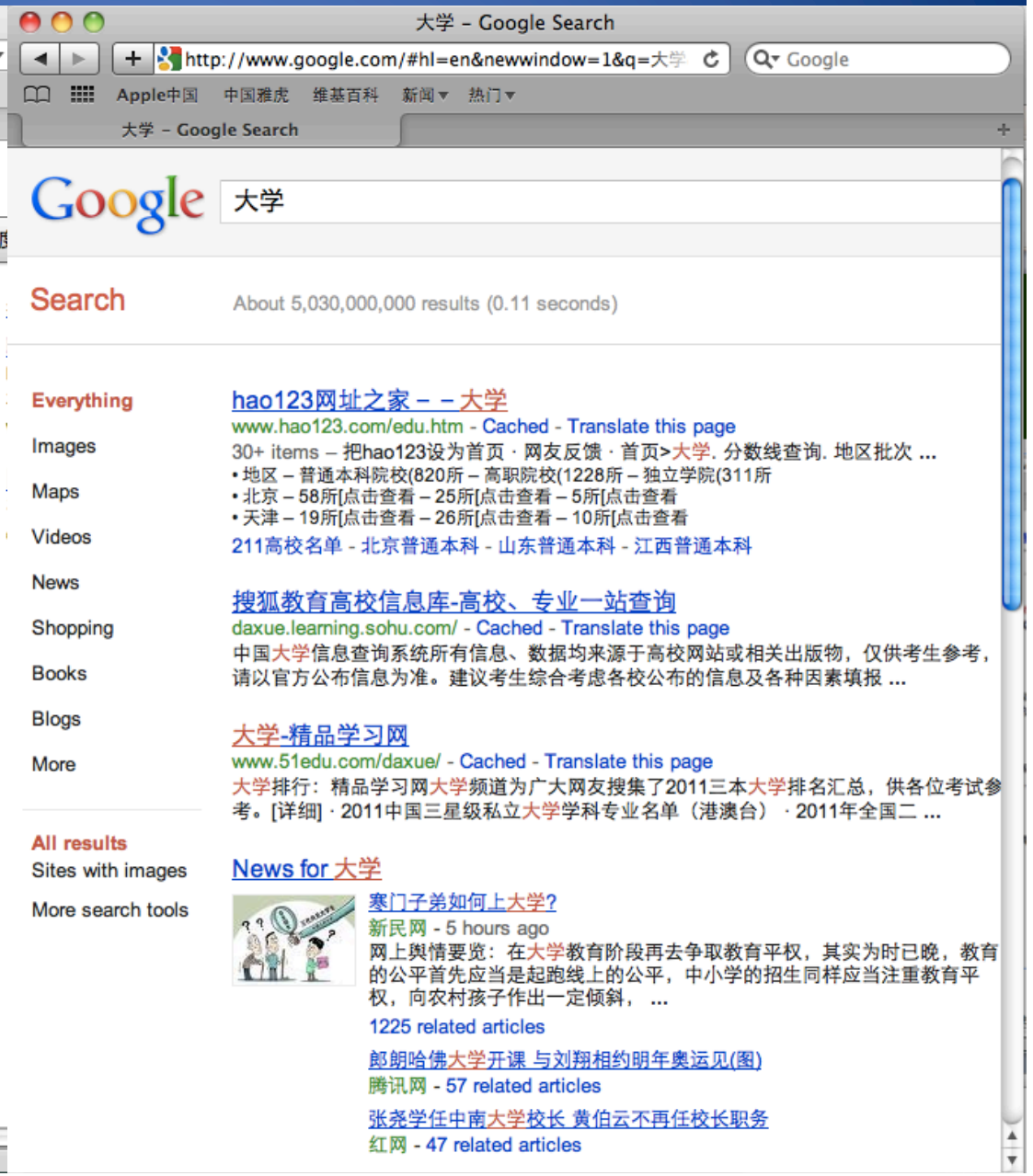
- 但人们在网络上不光是要找“文献”，而是多方面意义的“信息”
- 例如，人们给出“北京航空航天大学”查询词，多数会有什么预期？
- 查询“大学”呢？（意图会相当多样化）



为什么能恰到好处？

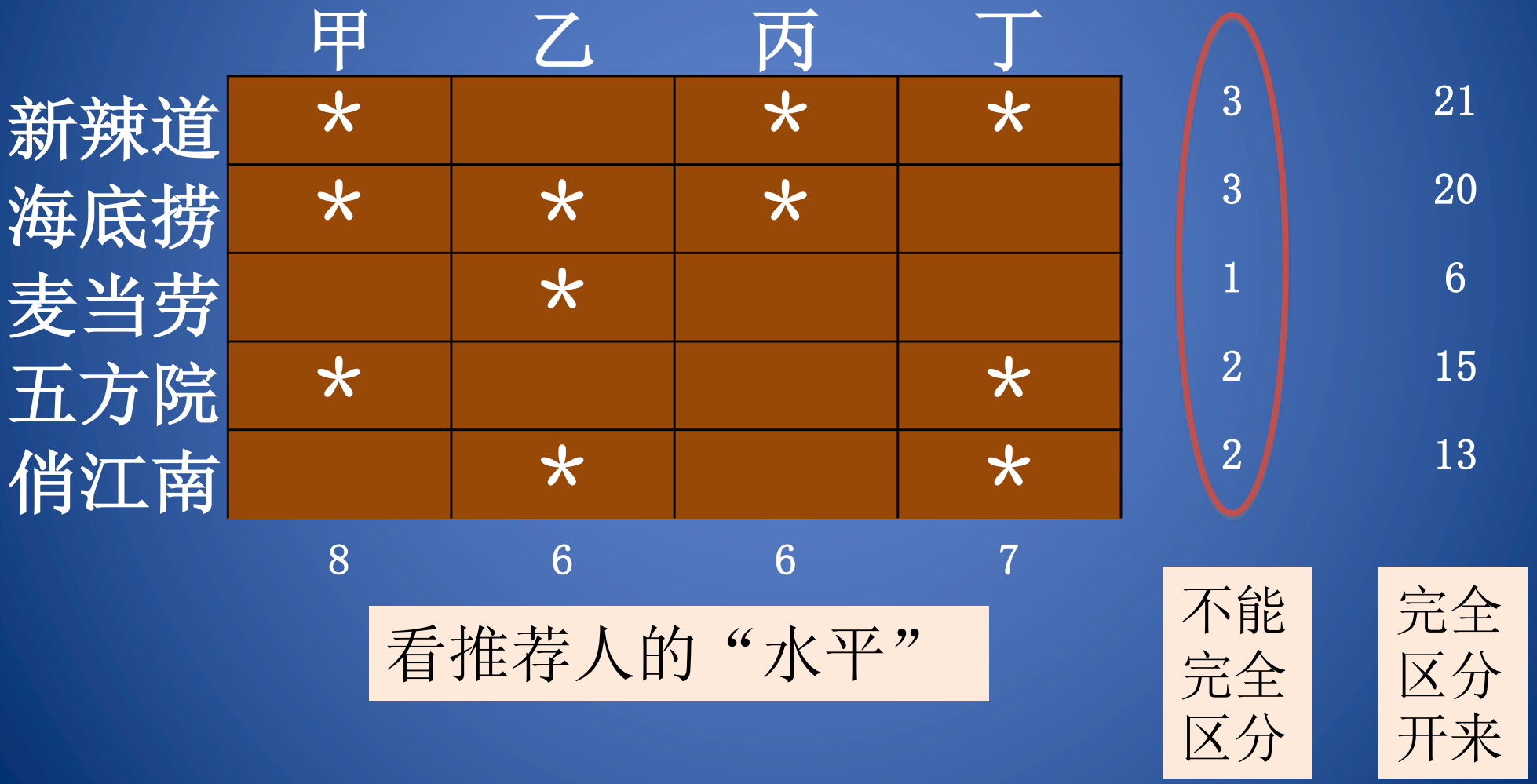
- 主页放在最前面，一定不是因为其中包含许多“北京航空航天大学”字样
- 很可能是由于许多包含“北京航空航天大学”字样的网页指向它
  - 利用链接中隐含的信息

这个两个结果哪一个较好？



有效利用链接关系蕴含的信息，是搜索引擎超越传统信息检索系统、技术进步的最重要标志

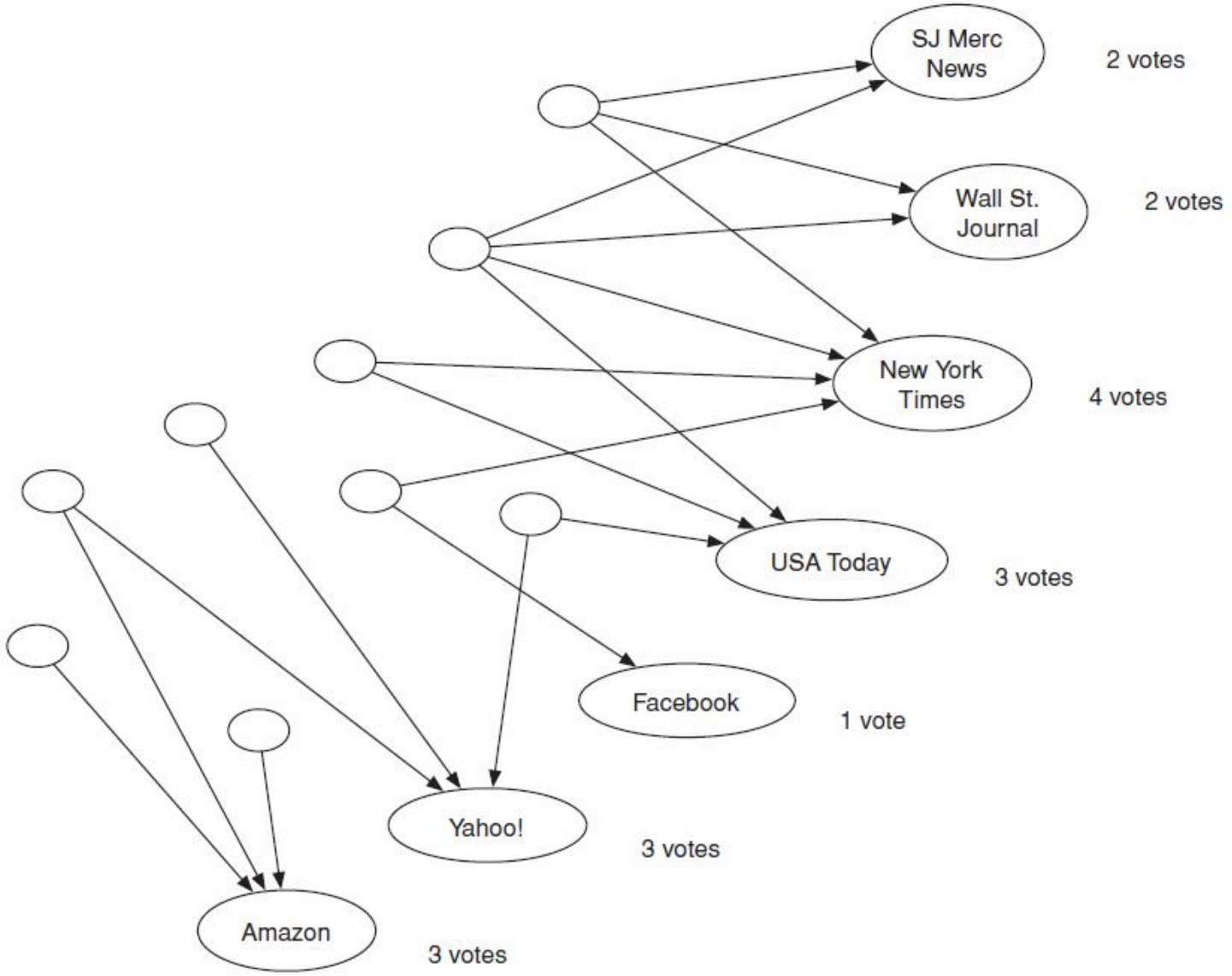
餐馆推荐问题


甲
乙
丙
丁
新辣道 * * *
海底捞 * * *
麦当劳 *
五方院 * *
俏江南 * *
3
3
1
2
2
21
20
6
15
13
8
6
6
7
看推荐人的“水平”
不能完全区分
完全区分开来

反复改进原理（例）

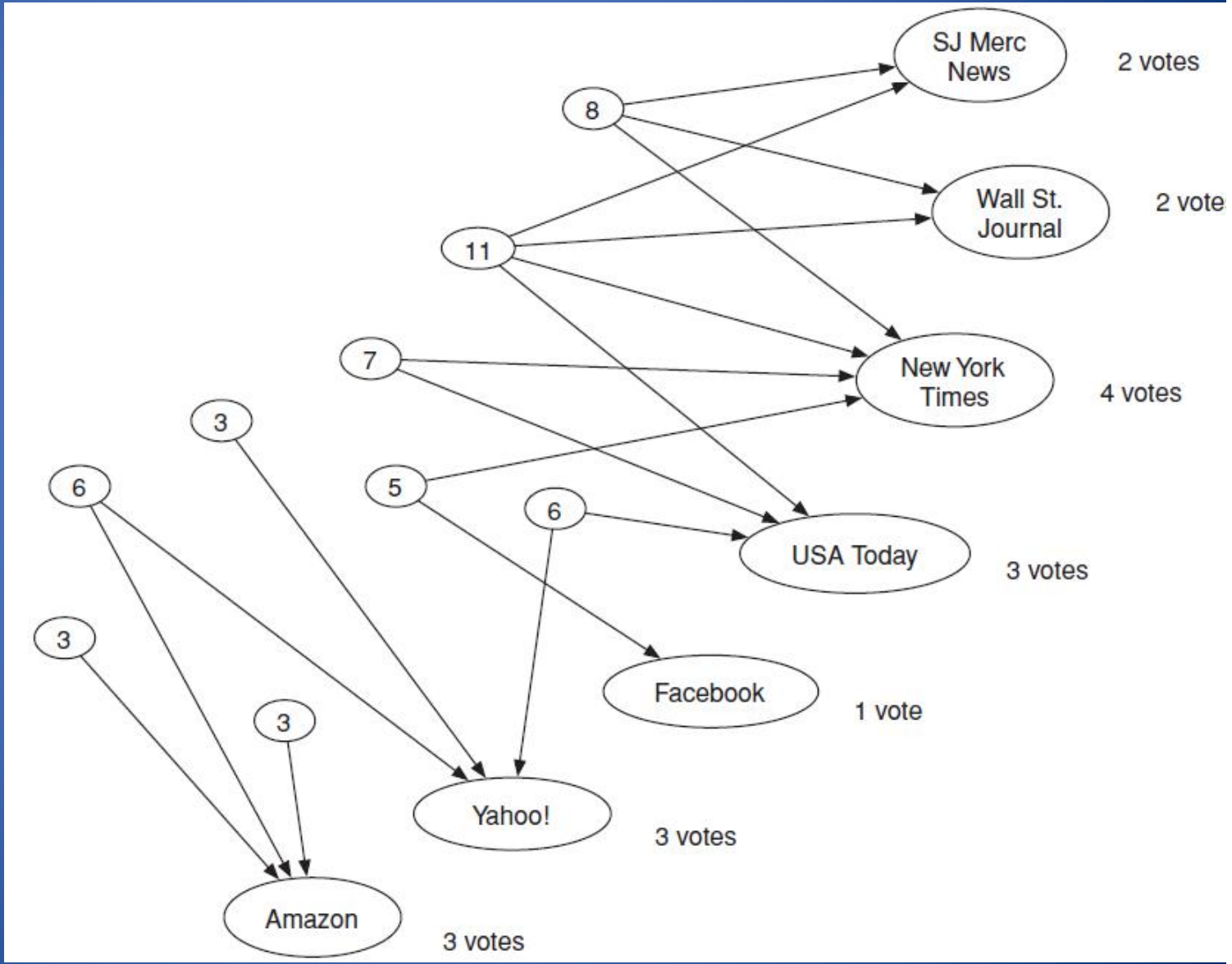
假设查询词“newspaper”

- 左边是与“newspaper”字面上相关的网页。
- 右边是它们所指向的网页，得到的“票数”表示一定的认可度

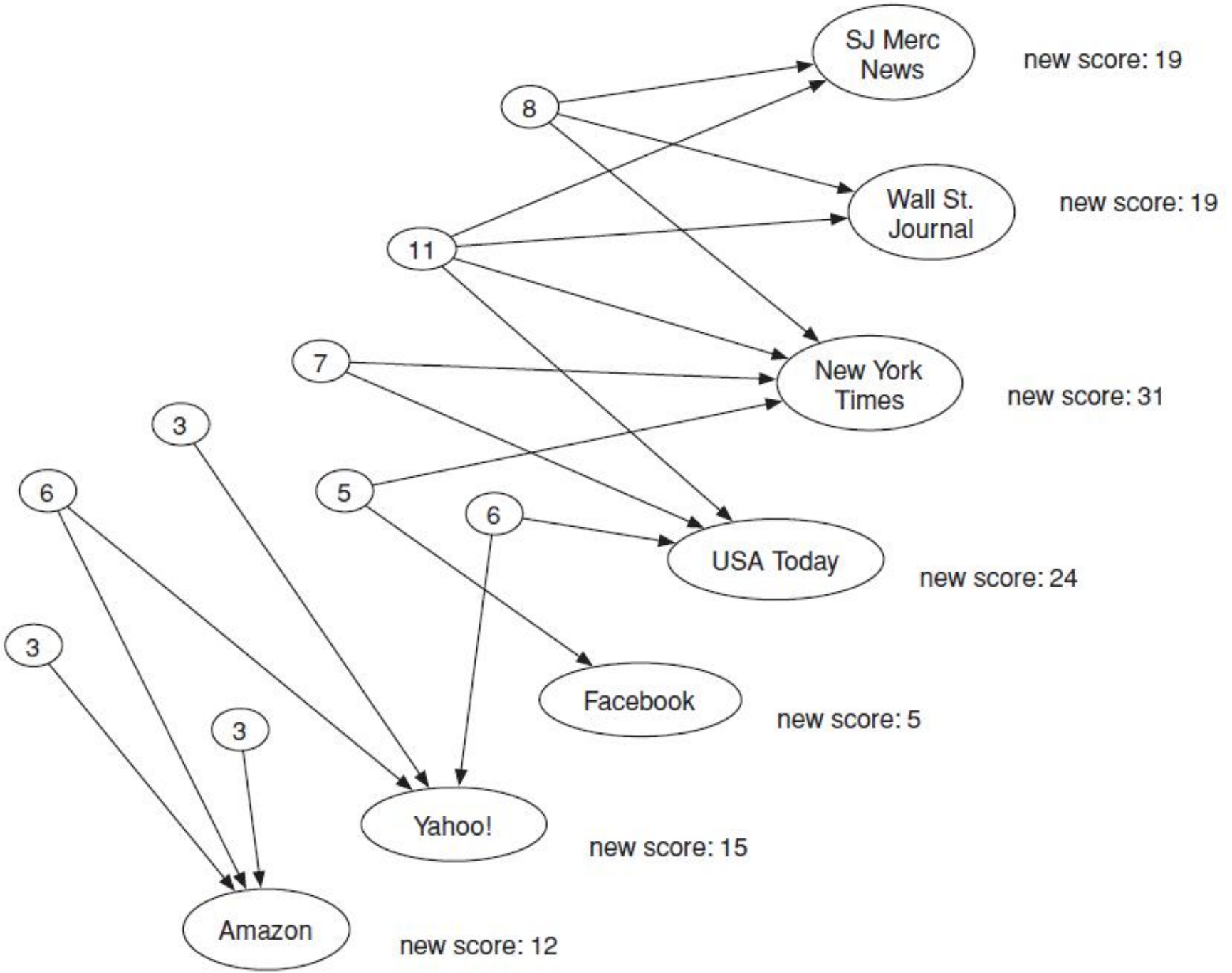


反复改进原理（续）
（principle of repeated improvement）

- 也可以反过来评估“推荐者”的份量
- 然后可以在考虑推荐者份量的情况下重新评估网站相对于“newspaper”的重要性



反复改进原理


SJ Merc News
new score: 19
8
Wall St. Journal
new score: 19
11
7
New York Times
new score: 31
3
6
5
6
USA Today
new score: 24
3
Facebook
new score: 5
3
Yahoo!
new score: 15
Amazon
new score: 12

“中枢”（hub）与“权威”（authority）

- 万维网中一篇网页的两面属性。观念：
  - 被很多网页指向：权威性高
  - 指向很多网页：中枢性强
- **HITS算法**：计算网页的权威值（**auth**）和中枢值（**hub**）
  - **Hyperlink-Induced Topic Search**
- 在实际中算法实施的针对性：相关网页集合
  - 不是全部网页集合。为避免赘述，这一点后面不再总强调，认为所讨论的就是相关网页集合。

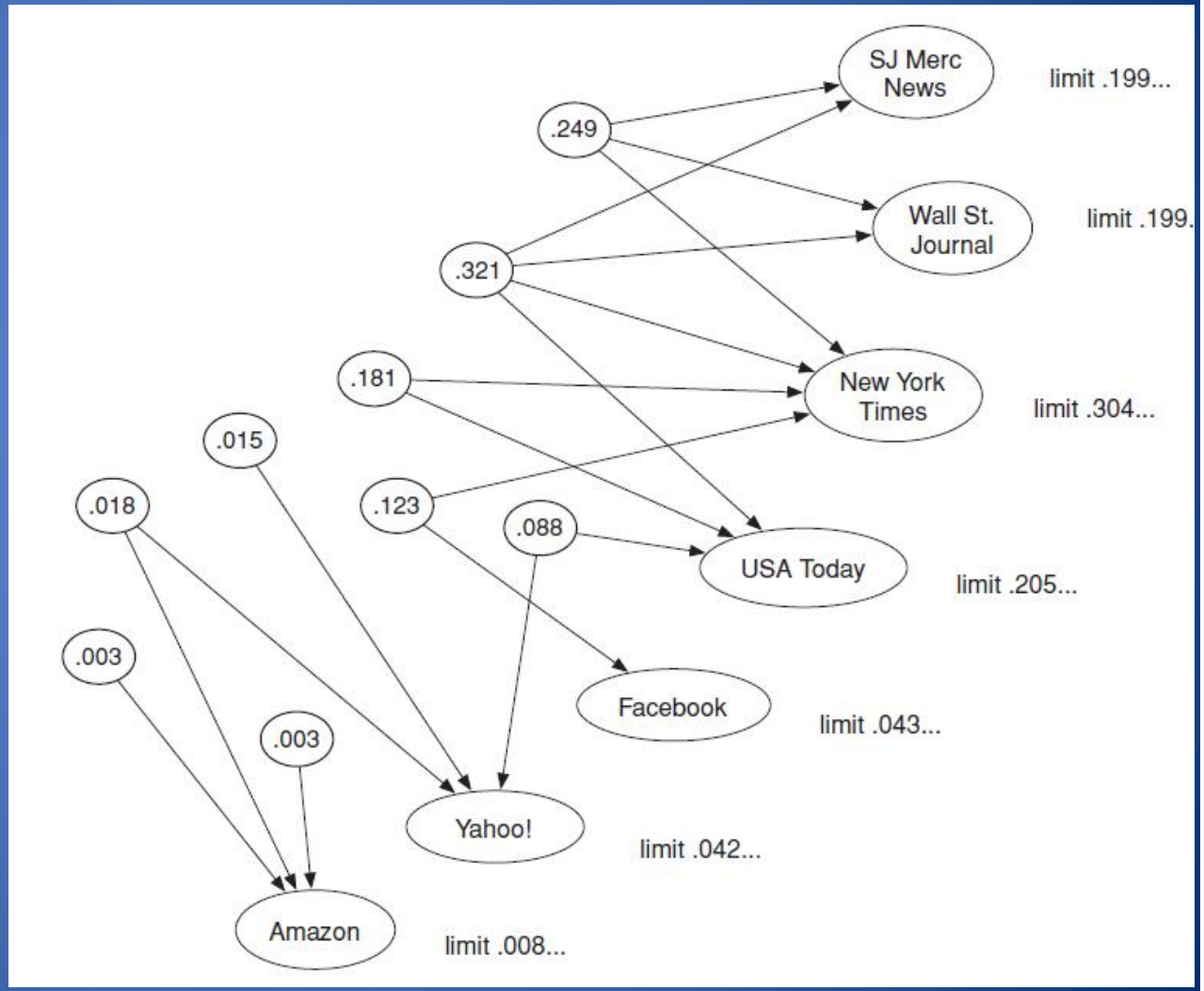
auth(p) 和 hub(p) 的计算方法

- 输入：一个有向图
- 初始化：对于每一个节点p，auth(p)=1，hub(p)=1
- 利用中枢值更新权威值
  - 对于每一个节点p，让auth(p)等于指向p的所有节点q的hub(q)之和
- 利用权威值更新中枢值
  - 对于每一个节点p，让hub(p)等于p指向的所有节点q的auth(q)之和
- 重复上述两步若干（k）次

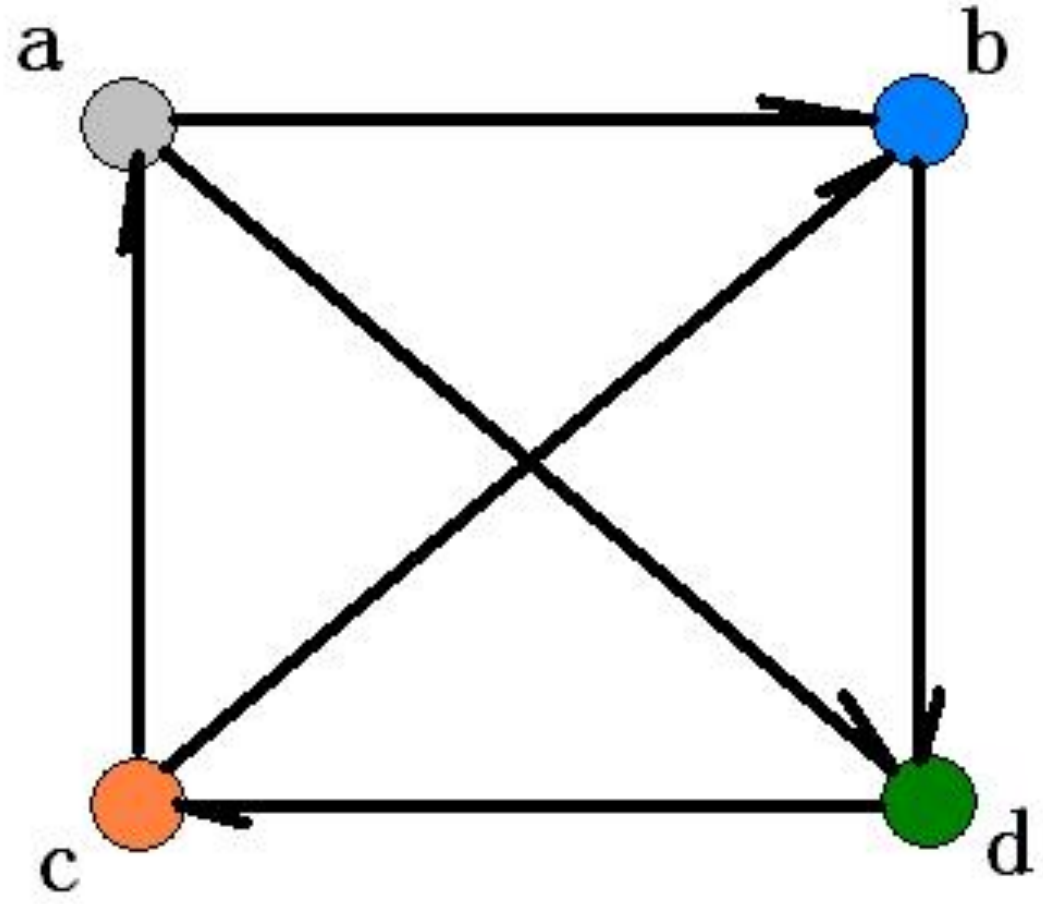
在搜索引擎领域，auth值或hub值高的网页，有时分别称为“权威网页”和“中枢网页”。一篇网页可以兼具二者。

归一化与极限

- 数值随迭代次数递增
- Auth和hub值的意义在于相对大小
- 在每一轮结束后做归一化：值 / 总和
- 归一化结果随迭代次数趋向于一个极限
  - 相继两次迭代的值不变
  - 极限与初值无关，即存在“均衡”



PageRank：节点的一种重要性测度



$$a=c/2$$
$$b=a/2+c/2$$
$$c=d$$
$$d=a/2+b$$

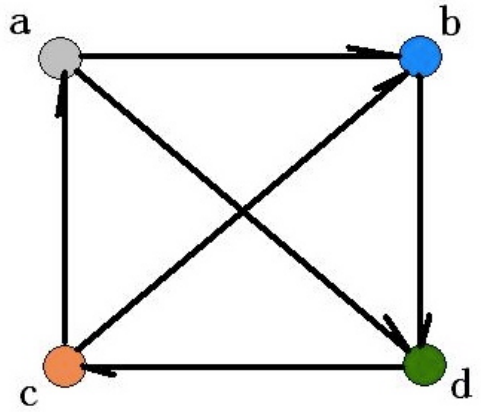
搜索引擎形成查询结果网页排序的重要参数

上图的算例

经过约**70**次迭代，最后收敛到：
A=0.615, B=0.923, C=D=1.231

E14

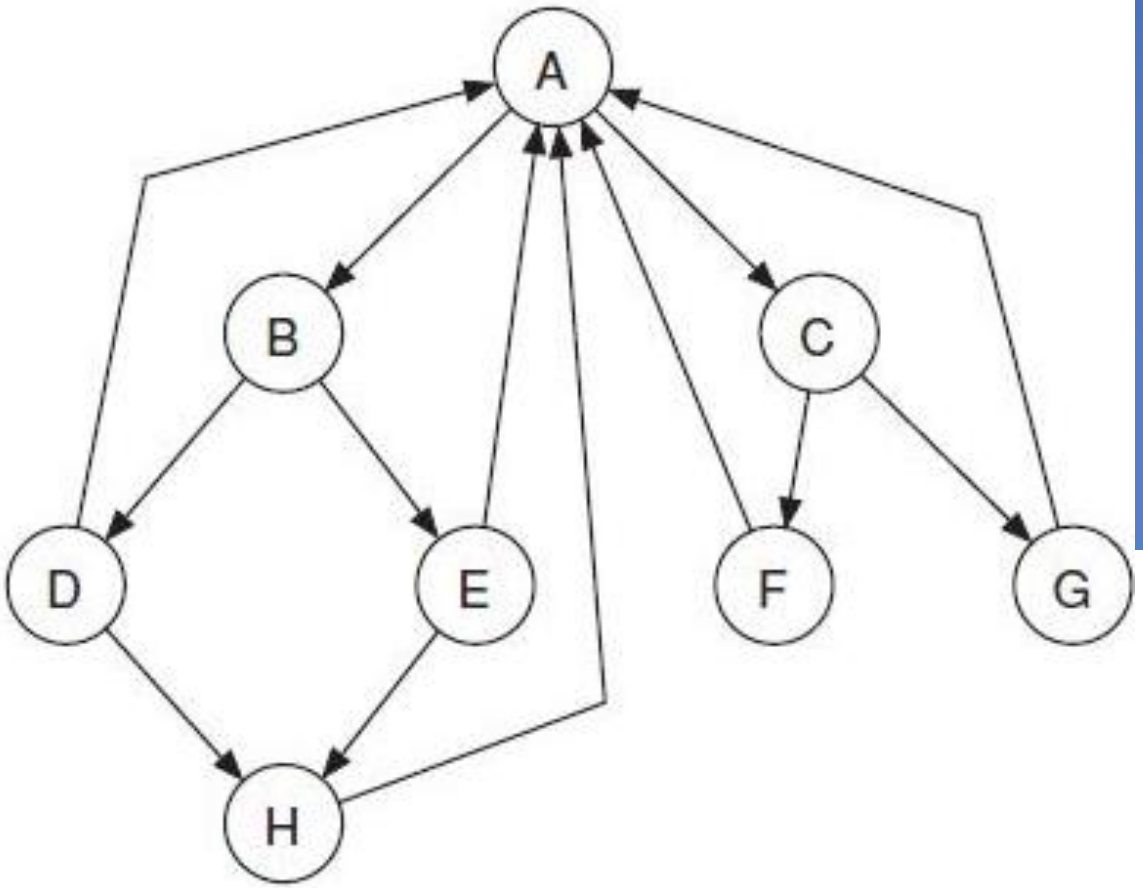
	A	B	C	D	E
1	a=c/2	b=a/2+c/2	c=d	d=a/2+b	传递关系
2	a	b	c	d	
3	1	1	1	1	初值
4	0.5	1	1	1.5	按传递关系计算
5	0.5	0.75	1.5	1.25	
6	0.75	1	1.25	1	
7	0.625	1	1	1.375	
8	0.5	0.8125	1.375	1.3125	
9	0.6875	0.9375	1.3125	1.0625	
10	0.65625	1	1.0625	1.28125	
11	0.53125	0.859375	1.28125	1.328125	
12	0.640625	0.90625	1.328125	1.125	
13	0.6640625	0.984375	1.125	1.2265625	



PageRank基本算法描述

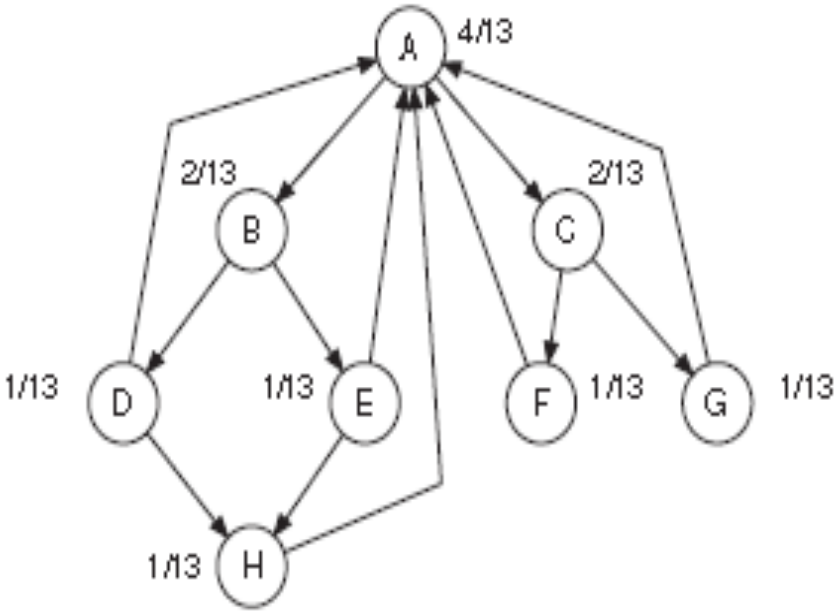
- 输入：一个有n个节点的网络（有向图），设所有节点的PageRank初始值为1/ n。
- 选择操作的步骤数k
- 对PageRank做k次更新操作，每次使用以下规则：
  - 每个节点将自己当前的PageRank值通过出向链接均分传递给所指向的节点
    - 若没有出向链接，则认为传递给自己
  - 每个节点以从入向链接获得的（包括可能自传的）所有值之和更新它的PageRank

一个计算网页排名的实例



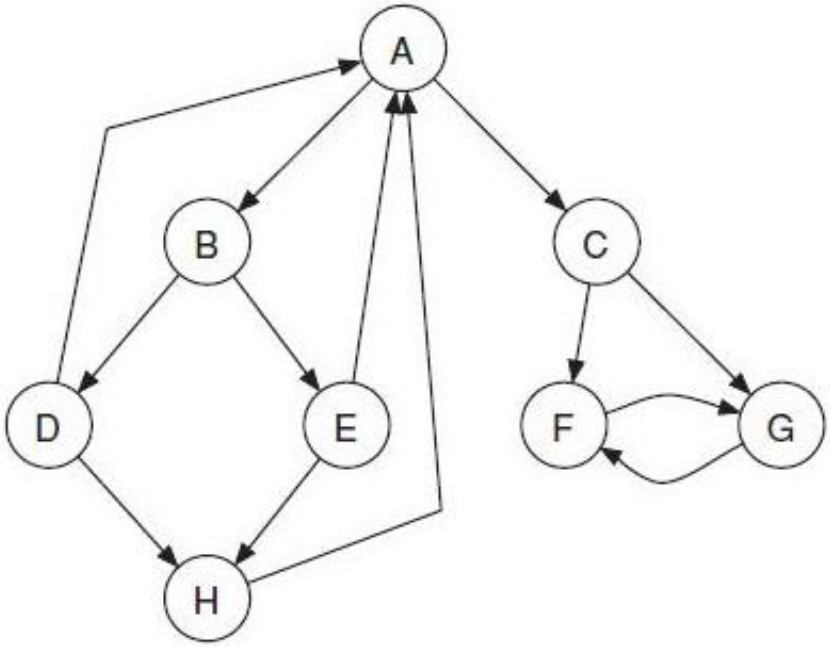
- 每个节点的初值都是1/8
- 最后收敛结果见下图

Step	A	B	C	D	E	F	G	H
1	$\frac{1}{2}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$
2	$\frac{3}{16}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$



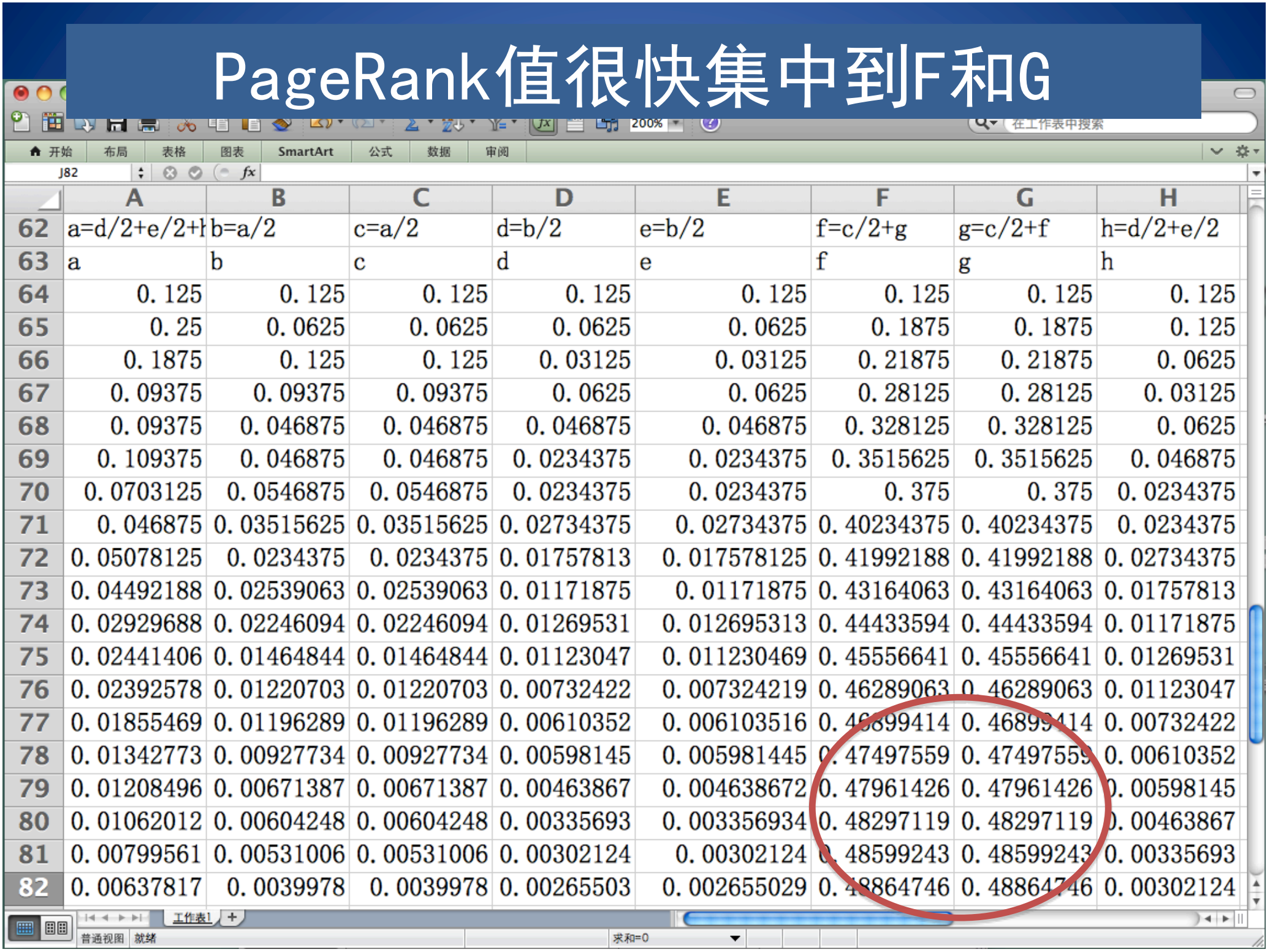
PageRank基本算法在某些网络结构上表现不好

- PageRank算法不象HITS算法那样需要归一化问题，但有新问题
- F和G两个节点显得很“自私”：吸收别人的价值，但不向外传
  - 导致它们最后各自1/2，其他人都0



这也显示了共谋（colluding）制造垃圾网页的一个原理

PageRank值很快集中到F和G

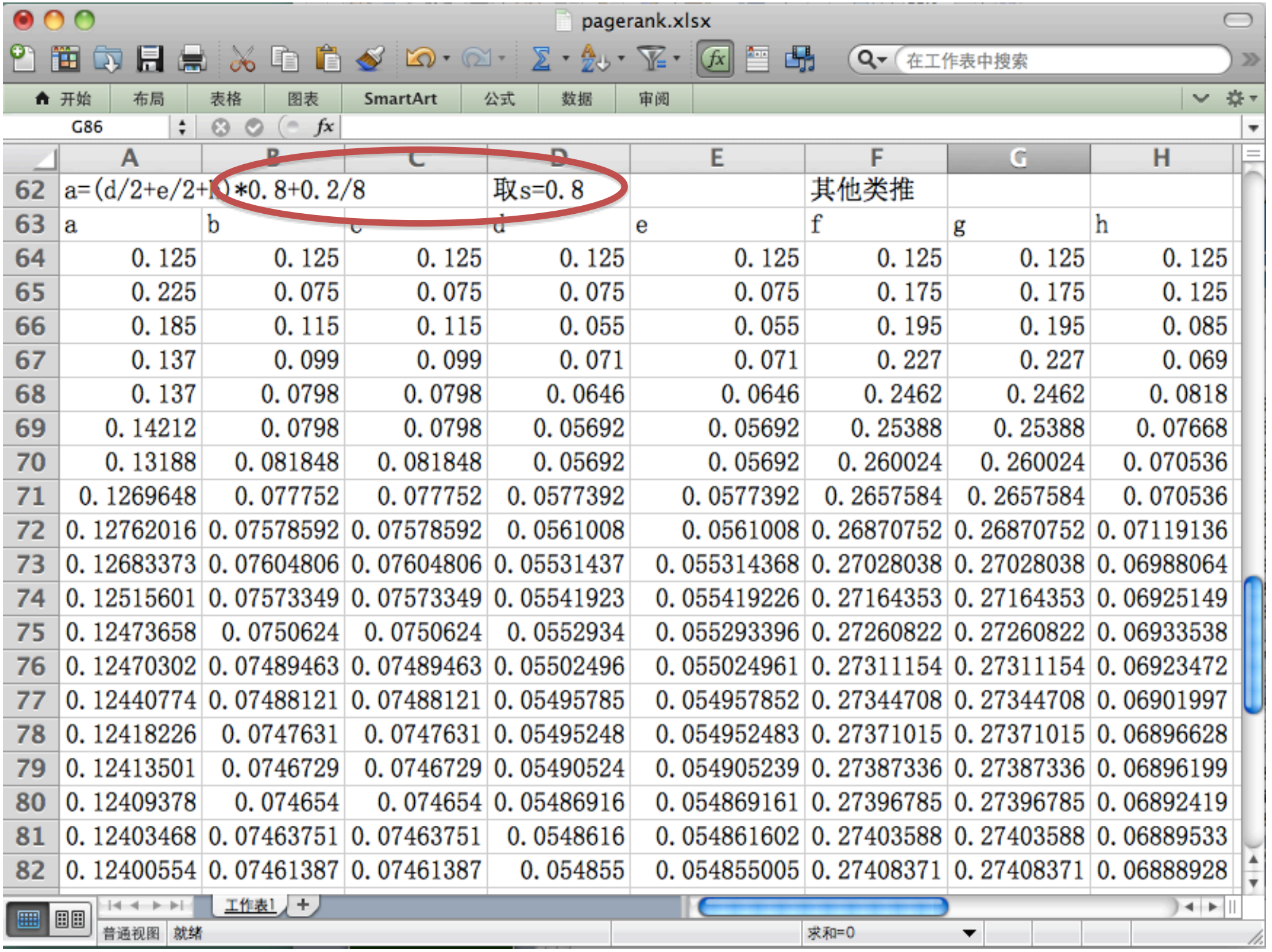


	A	B	C	D	E	F	G	H
62	a=d/2+e/2+h	b=a/2	c=a/2	d=b/2	e=b/2	f=c/2+g	g=c/2+f	h=d/2+e/2
63	a	b	c	d	e	f	g	h
64	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
65	0.25	0.0625	0.0625	0.0625	0.0625	0.1875	0.1875	0.125
66	0.1875	0.125	0.125	0.03125	0.03125	0.21875	0.21875	0.0625
67	0.09375	0.09375	0.09375	0.0625	0.0625	0.28125	0.28125	0.03125
68	0.09375	0.046875	0.046875	0.046875	0.046875	0.328125	0.328125	0.0625
69	0.109375	0.046875	0.046875	0.0234375	0.0234375	0.3515625	0.3515625	0.046875
70	0.0703125	0.0546875	0.0546875	0.0234375	0.0234375	0.375	0.375	0.0234375
71	0.046875	0.03515625	0.03515625	0.02734375	0.02734375	0.40234375	0.40234375	0.0234375
72	0.05078125	0.0234375	0.0234375	0.01757813	0.017578125	0.41992188	0.41992188	0.02734375
73	0.04492188	0.02539063	0.02539063	0.01171875	0.01171875	0.43164063	0.43164063	0.01757813
74	0.02929688	0.02246094	0.02246094	0.01269531	0.012695313	0.44433594	0.44433594	0.01171875
75	0.02441406	0.01464844	0.01464844	0.01123047	0.011230469	0.45556641	0.45556641	0.01269531
76	0.02392578	0.01220703	0.01220703	0.00732422	0.007324219	0.46289063	0.46289063	0.01123047
77	0.01855469	0.01196289	0.01196289	0.00610352	0.006103516	0.46899414	0.46899414	0.00732422
78	0.01342773	0.00927734	0.00927734	0.00598145	0.005981445	0.47497559	0.47497559	0.00610352
79	0.01208496	0.00671387	0.00671387	0.00463867	0.004638672	0.47961426	0.47961426	0.00598145
80	0.01062012	0.00604248	0.00604248	0.00335693	0.003356934	0.48297119	0.48297119	0.00463867
81	0.00799561	0.00531006	0.00531006	0.00302124	0.00302124	0.48599243	0.48599243	0.00335693
82	0.00637817	0.0039978	0.0039978	0.00265503	0.002655029	0.48864746	0.48864746	0.00302124

PageRank的同比缩减与统一补偿规则

- 同比缩减
  - 在每次运行基本PageRank更新规则后，将每一节点的PageRank值都乘以一个比例因子s，0<s<1，经验值在0.8-0.9之间。
- 统一补偿
  - 在每一节点的PageRank值上统一加上(1-s)/n。

这样，既维持了所有PageRank之和等于1的性质，也防止了PageRank值不恰当地集中到个别节点。



	A	B	C	D	E	F	G	H
62	a=(d/2+e/2+f)*0.8+0.2/8			取s=0.8		其他类推		
63	a	b	c	d	e	f	g	h
64	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
65	0.225	0.075	0.075	0.075	0.075	0.175	0.175	0.125
66	0.185	0.115	0.115	0.055	0.055	0.195	0.195	0.085
67	0.137	0.099	0.099	0.071	0.071	0.227	0.227	0.069
68	0.137	0.0798	0.0798	0.0646	0.0646	0.2462	0.2462	0.0818
69	0.14212	0.0798	0.0798	0.05692	0.05692	0.25388	0.25388	0.07668
70	0.13188	0.081848	0.081848	0.05692	0.05692	0.260024	0.260024	0.070536
71	0.1269648	0.077752	0.077752	0.0577392	0.0577392	0.2657584	0.2657584	0.070536
72	0.12762016	0.07578592	0.07578592	0.0561008	0.0561008	0.26870752	0.26870752	0.07119136
73	0.12683373	0.07604806	0.07604806	0.05531437	0.055314368	0.27028038	0.27028038	0.06988064
74	0.12515601	0.07573349	0.07573349	0.05541923	0.055419226	0.27164353	0.27164353	0.06925149
75	0.12473658	0.0750624	0.0750624	0.0552934	0.055293396	0.27260822	0.27260822	0.06933538
76	0.12470302	0.07489463	0.07489463	0.05502496	0.055024961	0.27311154	0.27311154	0.06923472
77	0.12440774	0.07488121	0.07488121	0.05495785	0.054957852	0.27344708	0.27344708	0.06901997
78	0.12418226	0.0747631	0.0747631	0.05495248	0.054952483	0.27371015	0.27371015	0.06896628
79	0.12413501	0.0746729	0.0746729	0.05490524	0.054905239	0.27387336	0.27387336	0.06896199
80	0.12409378	0.074654	0.074654	0.05486916	0.054869161	0.27396785	0.27396785	0.06892419
81	0.12403468	0.07463751	0.07463751	0.0548616	0.054861602	0.27403588	0.27403588	0.06889533
82	0.12400554	0.07461387	0.07461387	0.054855	0.054855005	0.27408371	0.27408371	0.06888928

随机游走：PageRank的另一种等价理解

- 想象一个人从一篇随机选择的网页开始，随机选择其中的链接浏览到下一篇网页，并不断如此进行，称为“随机游走”。
- 考虑一篇网页X，问：经过k步随机游走到达X的概率是多少？
- 可以证明：到达X的概率等于运行PageRank基本算法k步得到的值。
- 随机游走概念稍加修改也可以和同比缩减统一补偿的PageRank等价。

链接分析小结

- 有向图作为万维网的结构模型
  - 概貌形状：领结
  - 得到领结概貌形状的方法：宽度优先搜索
- 万维网信息节点的（结构性）重要程度
  - 权威性，中枢性：一个节点的双重作用
    - 反复改进原理（交叉支持原理）
    - 相对查询需求或者某个主题
  - **PageRank**（网页排名）：节点的全局相对重要性
    - 局部结构信息在全局扩散达到的均衡

作业

- 第14章 2,4