第10讲 受限资源约束下的算法—排序问题求解示例

黄宏

华中科技大学计算机学院

honghuang@hust.edu.cn

第10讲-受限资源约束下的算法—排序问题求解示例

一、为什么要研究排序算法

二、内排序基本算法

三、PageRank算法--网页排序

为什么要研究排序算法

(1)什么是排序问题?

对一组对象按照某种规则进行有序排列。通常是把一组对象整理成按关键字递增(或递减)的排列，关键字是指对象的一个用于排序的特性。

例如：

- 对一组“人”，按“年龄”或“身高”排序；
- 对一组“商品”，按“价格”排序；
- 对一组“网页”，按“重要度”排序；
- 对一组“词汇”，按“首字母”字典序排序。
- …… …

为什么要研究排序算法

(2)从结构化数据表查找看排序问题的重要性

未排序

学号	姓名	成绩
120300101	李鹏	88
120300105	张伟	66
120300107	闫宁	95
120300102	王刚	79
120300103	李宁	94
120300106	徐月	85
120300108	杜岩	44
120300104	赵凯	69
120300109	江海	77
120300110	周峰	73

数据表记录数: n

查找成绩为80分的所有同学?

【算法A：未排序数据查找算法】

Start of algorithm

Step 1. 从数据表的第1条记录开始，直到其最后一条记录为止，读取每一条记录，做Step 2。

Step 2. 对每一条记录，判断成绩是否等于给定的分数：如果是，则输出；如果不是，则不输出。

End of algorithm

算法效率：读取并处理所有记录，即n条记录

为什么要研究排序算法

(2)从结构化数据表查找看排序问题的重要性

已排序

学号	姓名	成绩
120300107	闫宁	95
120300103	李宁	94
120300101	李鹏	88
120300106	徐月	85
120300102	王刚	79
120300109	江海	77
120300110	周峰	73
120300104	赵凯	69
120300105	张伟	66
120300108	杜岩	44

数据表记录数: n

查找成绩为80分的所有同学?

【算法B：已排序数据查找算法】

Start of algorithm

Step 1. 从数据表的第1条记录开始，直到其最后一条记录为止，读取每一条记录，做Step 2和Step 3步。

Step 2. 对每一条记录，判断成绩是否等于给定的分数：如果等于，则输出；如果不等于，则不输出。

Step 3. 判断该条记录的成绩是否小于给定的分数：如果不是，则继续；否则，退出循环，算法结束。

End of algorithm

算法效率：读取并处理部分记录，即<=n条记录

为什么要研究排序算法

(2)从结构化数据表查找看排序问题的重要性

已排序

学号	姓名	成绩
120300107	闫宁	95
120300103	李宁	94
120300101	李鹏	88
120300106	徐月	85
120300102	王刚	79
120300109	江海	77
120300110	周峰	73
120300104	赵凯	69
120300105	张伟	66
120300108	杜岩	44

数据表记录数: n

查找成绩为80分的所有同学?

【算法C：已排序数据查找算法】

Start of algorithm

Step 1. 假设数据表的最大记录数是n，待查询区间的起始记录位置Start为1，终止记录位置Finish为n；

Step 2. 计算中间记录位置I=(Start+Finish)/2，读取第I条记录。

Step 3. 判断第I条记录成绩是否大于给定查找分数：

(1)如果是小于，调整Finish = I-1, 如果Start >Finish则结束，否则继续做Step 2；(2)如果是大于，调整Start = I+1，如果Start>Finish则结束，否则继续做Step 2；(3)如果是等于，则输出，继续读取I周围所有的成绩与给定查找条件相等的记录并输出，直到所有相等记录查询输出完毕则算法结束。

End of algorithm

•算法效率：除极端情况外读取并处理<=n/2条记录

为什么要研究排序算法

(3)从结构化数据表的统计看排序问题的重要性

学号	姓名	成绩
120300107	闫宁	95
120300103	李宁	94
120300101	李鹏	88
120300106	徐月	85
120300102	王刚	79
120300109	江海	77
120300110	周峰	73
120300104	赵凯	69
120300105	张伟	66
120300108	杜岩	44

???

- 统计各个分数段的人数
- 统计每个同学的平均分数
- 统计每门课的平均分数

学号	姓名	成绩
120300101	李鹏	88
120300105	张伟	66
120300107	闫宁	95
120300102	王刚	79
120300103	李宁	94
120300106	徐月	85
120300108	杜岩	44
120300104	赵凯	69
120300109	江海	77
120300110	周峰	73

- 算法效率：需要读取并处理???条记录才能完成呢？

为什么要研究排序算法

(4)排序算法的重要性

学号	姓名	
120300107	闫宁	
120300103	李宁	
120300101	李鹏	88
120300106	徐月	
120300102	王刚	
120300109	江海	
120300110	周峰	
120300104	赵凯	
120300105	张伟	
120300108	杜岩	44

	名	成绩
	鹏	88
	伟	66
120300107	闫宁	95
	刚	79
	宁	94
	月	85
	岩	44
	凯	69
	海	77
120300110	周峰	73

排序算法是计算学科中很重要的算法

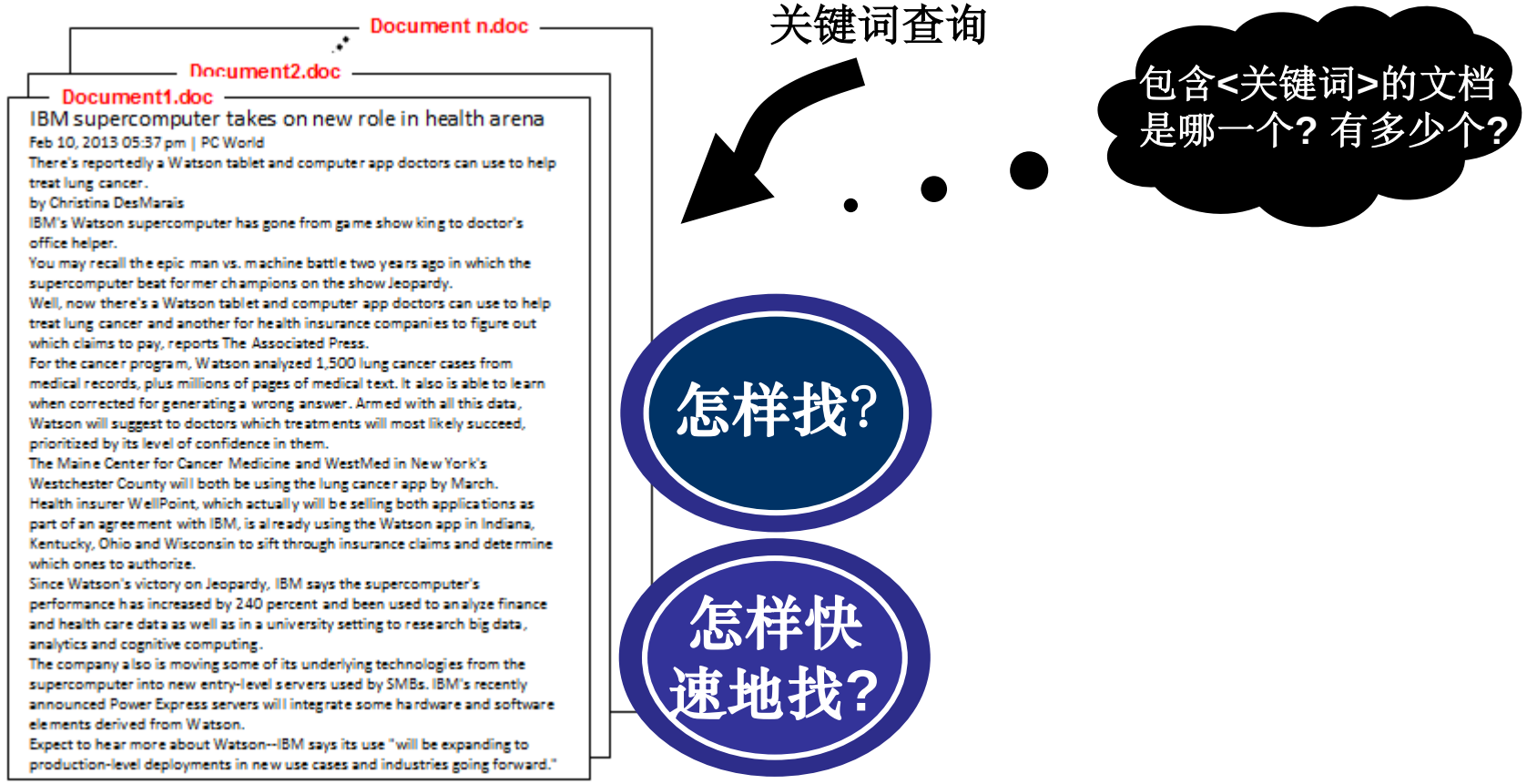
当对数据集合需要多遍处理时，先排序-好

排序算法是很多复杂算法的基础

•算法效率：需要读取并处理???条记录才能完成呢？

为什么要研究排序算法

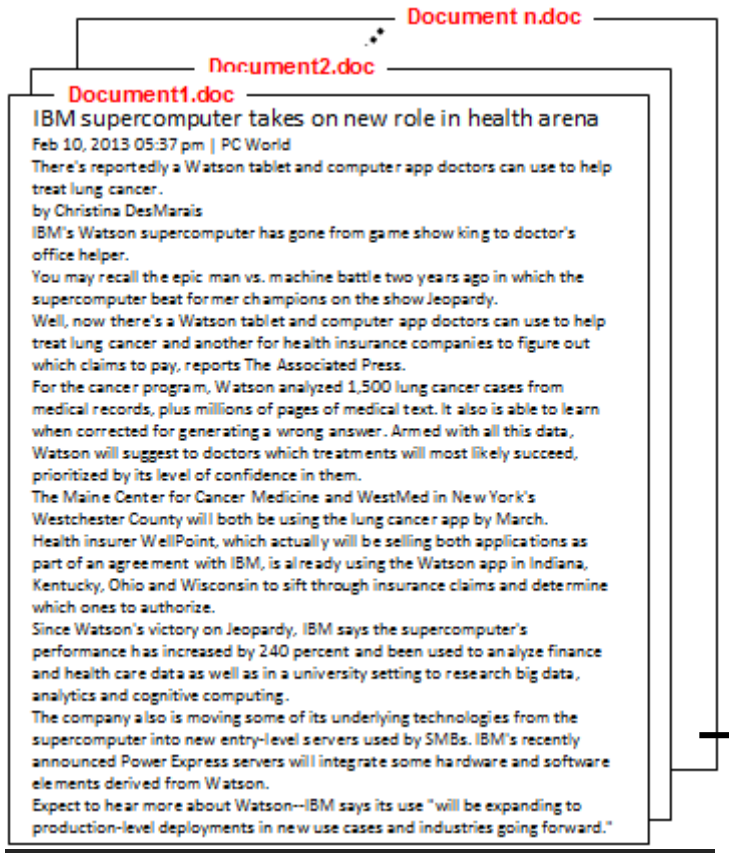
(5)非结构化文档检索



为什么要研究排序算法

(6)索引与倒排索引--需要排序?

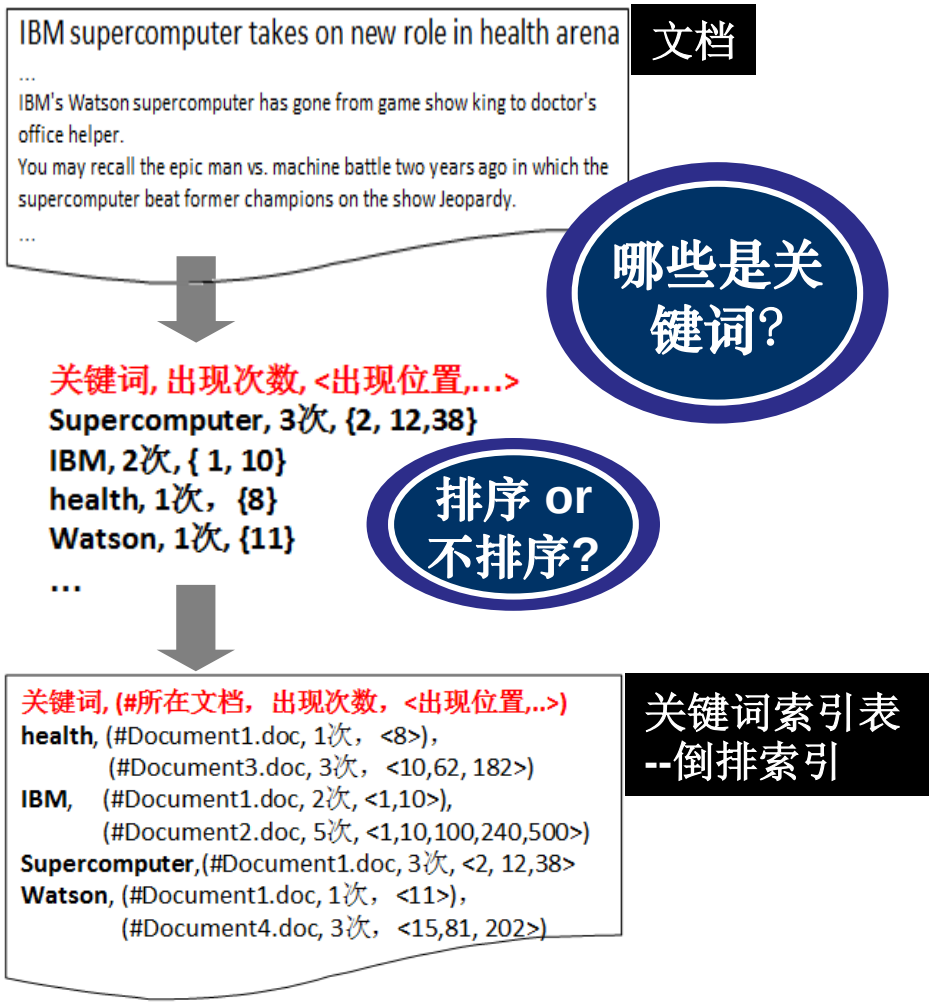
正排：一个文档包含了哪些词汇?
#Doc1, { Word1,Word2,... }
倒排：一个词汇包含在哪些文档中
Word1, { #Doc1, #Doc2, ... }



为什么要研究排序算法

(7)关键词的提取--需要排序?

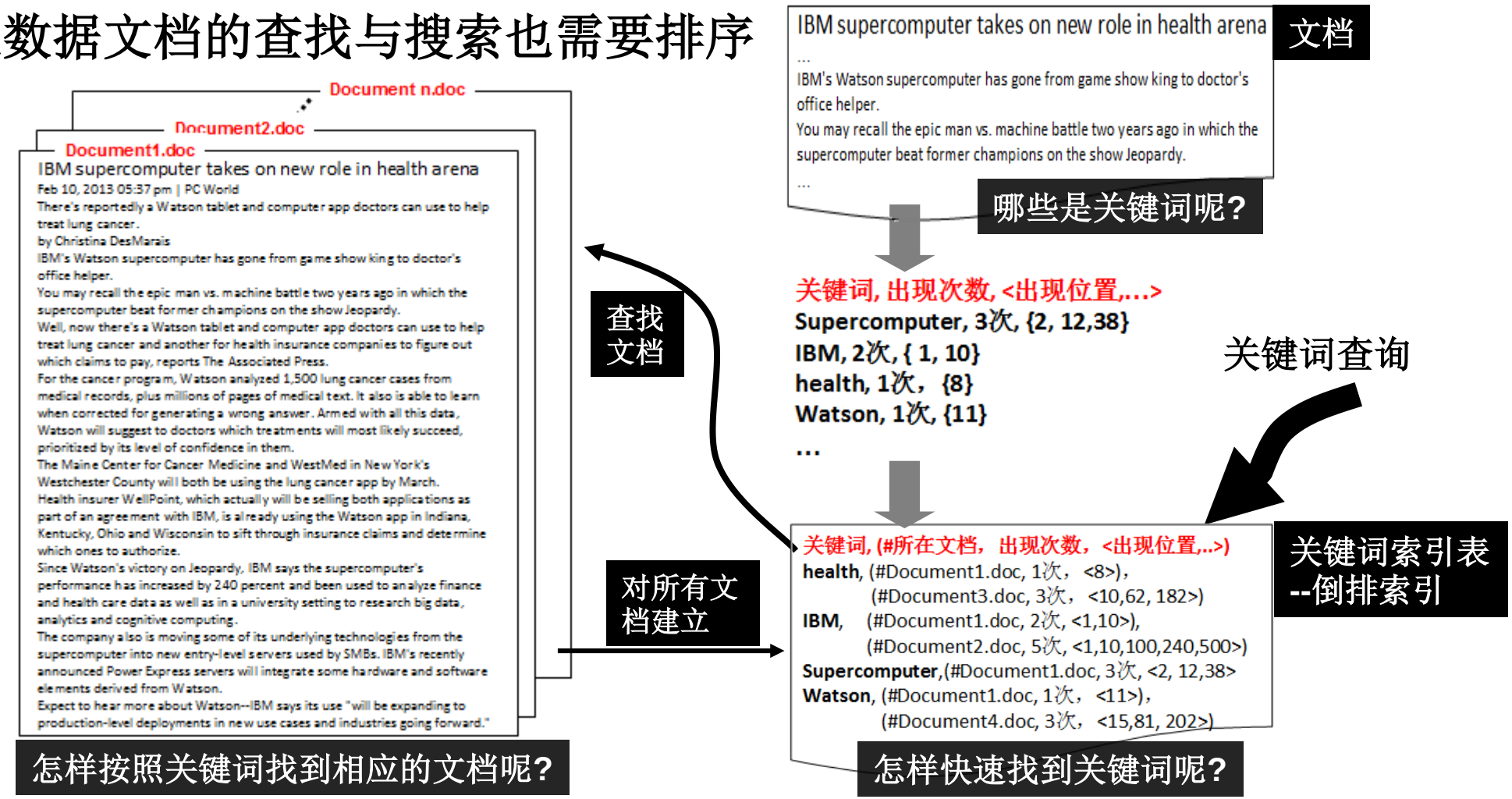
能否自动找出文档中的关键词?



为什么要研究排序算法

(8)从非结构化文档检索看排序问题的重要性

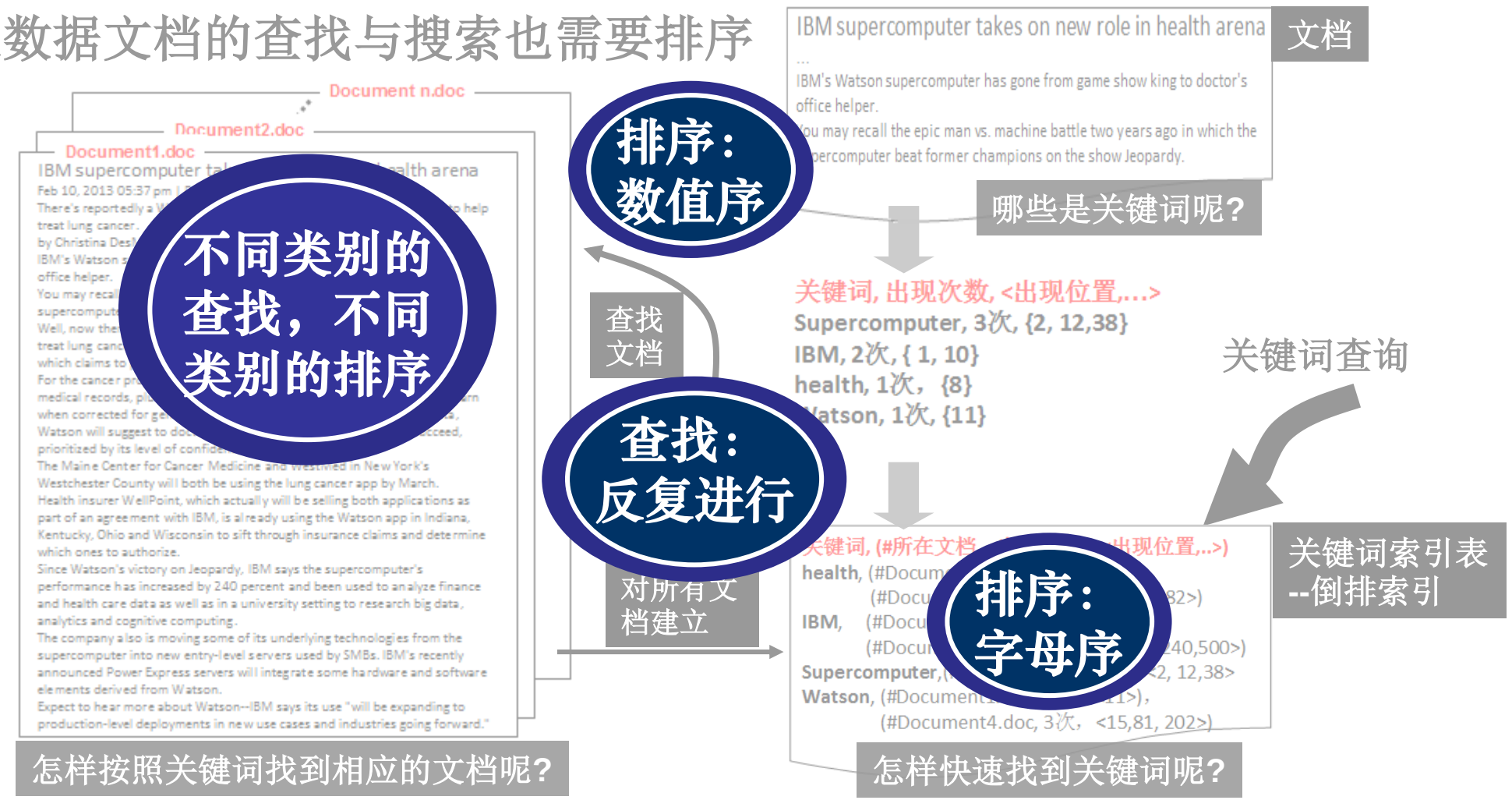
非结构化数据文档的查找与搜索也需要排序



为什么要研究排序算法

(8)从非结构化文档检索看排序问题的重要性

非结构化数据文档的查找与搜索也需要排序



第10讲-受限资源约束下的算法—排序问题求解示例

一、为什么要研究排序算法

二、内排序基本算法

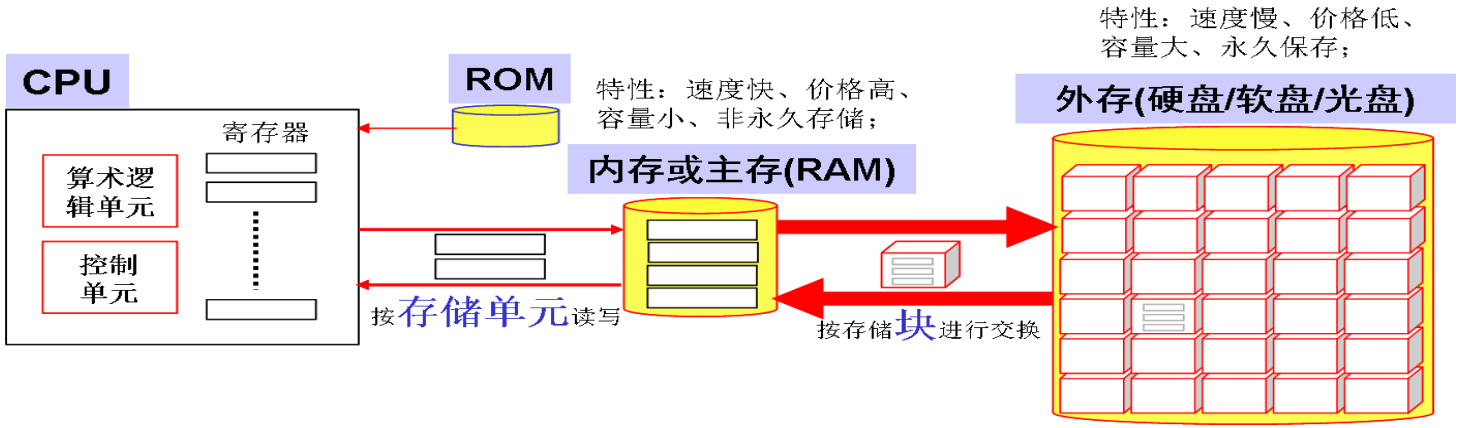
三、PageRank算法--网页排序

内排序基本算法

受限资源约束下的算法--内排序与外排序问题

●**内排序问题:**待排序的数据可一次性地装入内存中，即排序者可以完整地看到和操纵所有数据，使用数组或其他数据结构便可进行统一的排序处理的排序问题；

●**外排序问题:**待排序的数据保存在磁盘上，不能一次性装入内存，即排序者不能一次完整地看到和操纵所有数据，需要将数据分批装入内存分批处理的排序问题；



问题类比：某教师要对学生按身高排序。教师只能在房间(相当于内存)中对学生进行排序，假设房间仅能容纳100人，那么对于小于100人的学生排序便属于内排序问题。而对于大于100人，如1000人的学生排序，学生并不能都进入房间，而只能在操场(相当于磁盘)等候，轮流进入房间，这样的排序便属于外排序问题。

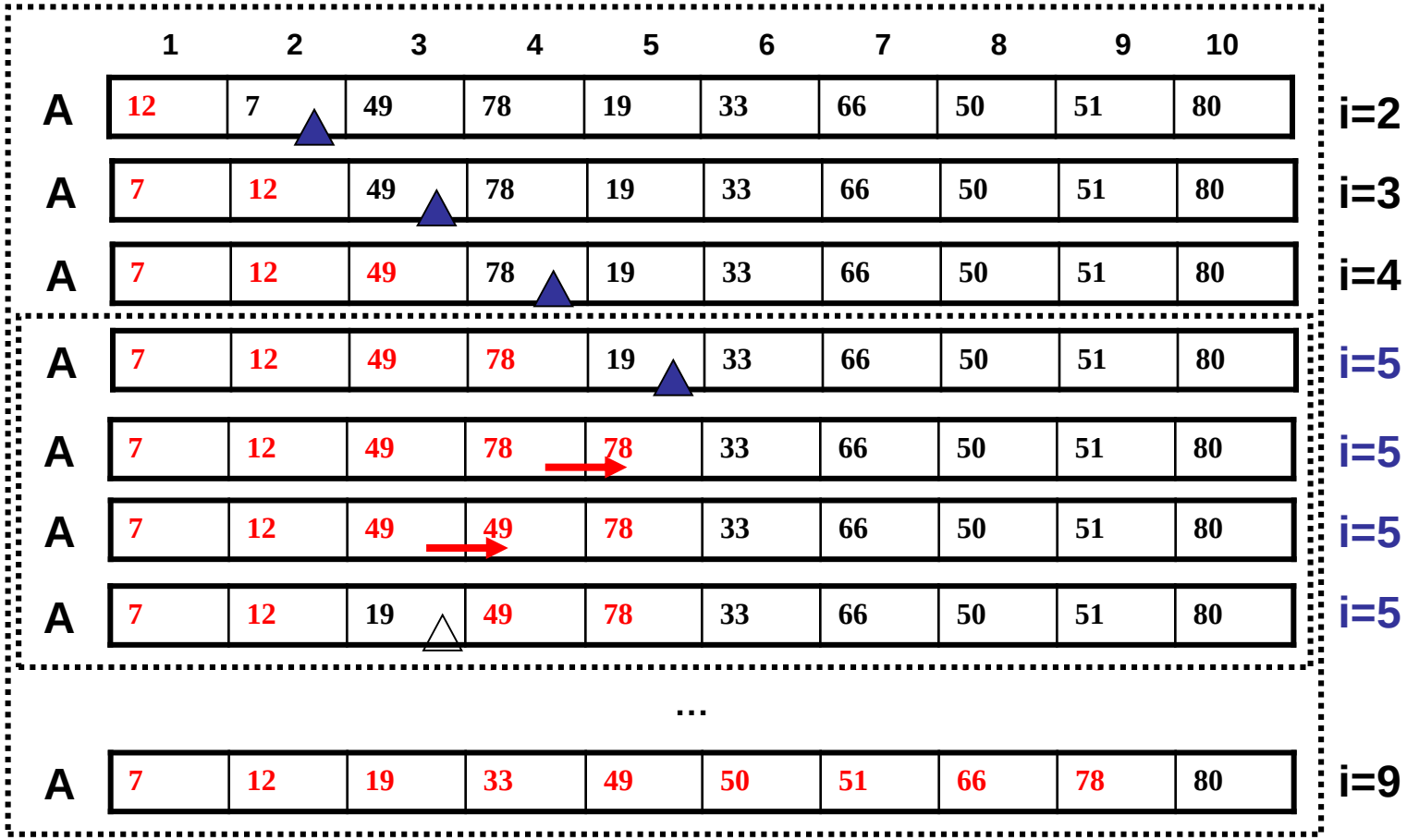
内排序基本算法

(1)【插入排序法】：基本思想

类似于打扑克牌时，一边抓牌，一边理牌的过程：
每抓一张牌就把它插入到适当的位置；
牌抓完了，也理完了。
---这种策略被称为插入排序

内排序基本算法

(1)【插入排序法】：过程模拟



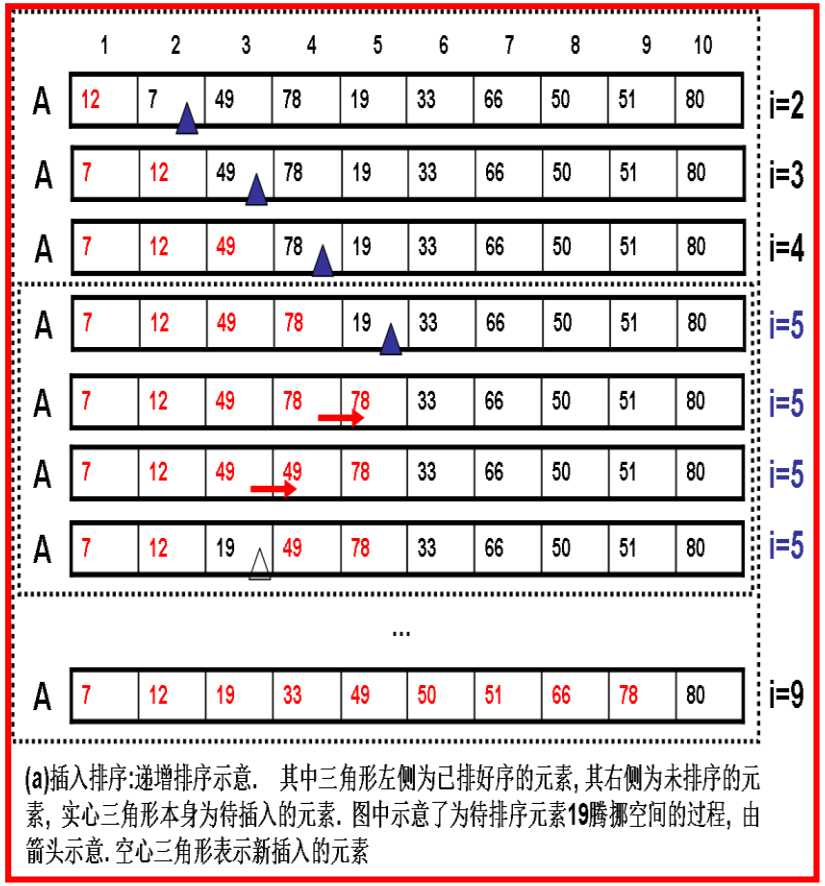
插入排序:递增排序示意.　其中三角形左侧为已排好序的元素, 其右侧为未排序的元素, 实心三角形本身为待插入的元素. 图中示意了为待排序元素19腾挪空间的过程, 由箭头示意. 空心三角形表示新插入的元素

内排序基本算法

(1)【插入排序法】：算法表达

```
INSERTION-SORT(A)
1.  for i=2 to N
2.  {    key = A[i] ;
3.           j =i-1;
4.           While (j>0 and A[j]>key)  do
5.           {  A[j+1]=A[j];
6.              j=j-1;  }
7.           A[j+1]=key;
8.  }
```

$O(N^2)$



(a)插入排序:递增排序示意. 其中三角形左侧为已排好序的元素,其右侧为未排序的元素, 实心三角形本身为待插入的元素. 图中示意了为待排序元素19腾挪空间的过程, 由箭头示意. 空心三角形表示新插入的元素

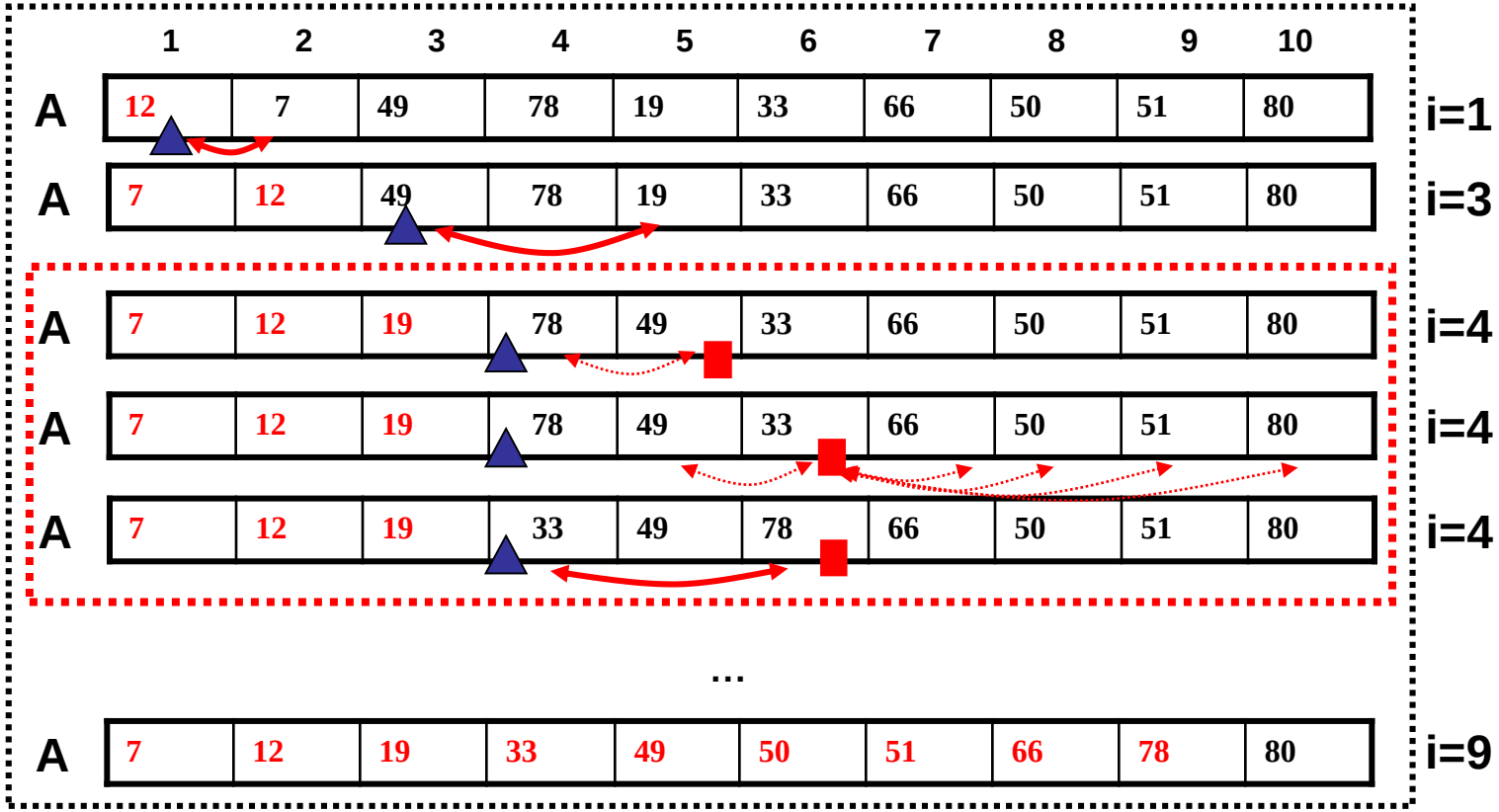
内排序基本算法

(2)【简单选择法】：基本思想

首先在所有数组元素中找出最小值的元素，放在A[1]中；
接着在不包含A[1]的余下数组元素中再找出最小值元素，放置在A[2]中；
如此下去，一直到最后一个元素。
这一排序策略被称为简单选择法排序。

内排序基本算法

(2)【简单选择法】：过程模拟



(b)选择排序:递增排序示意.

其中三角形代表本轮要找的最小元素应在的位置, 方形代表本轮次至今为止所找到的最小元素所在位置, 三角形左侧为已排好序的元素, 三角形右侧的每一元素依次和方形位置元素比较. 实线双向箭头代表两个元素交换. 虚线双向箭头代表两个元素需要比较

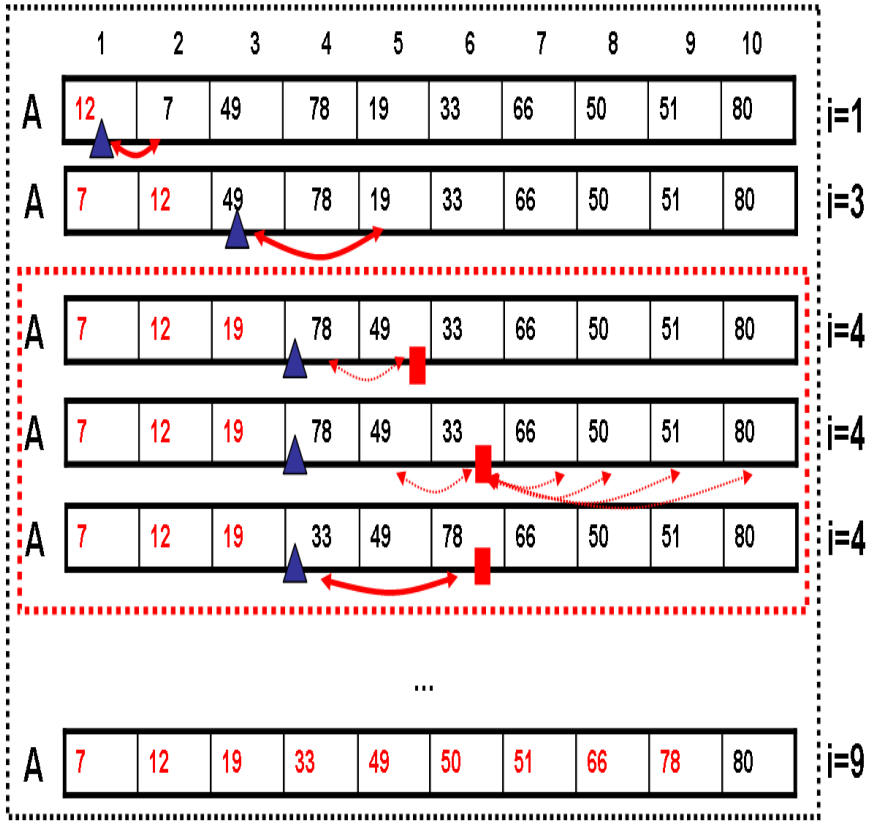
内排序基本算法

(2)【简单选择法】：算法表达

```
SELECTION-SORT(A)
1. for i=1 to N-1
2.  {       k=i;
3.          for j=i+1 to N
4.          { if A[j]<A[k] then k=j; }
5.          if k<>i then
6.          {
7.             temp =A[k];
8.            A[k]=A[i];
9.            A[i]=temp;
10.         }
11.  }
```

$O(N^2)$

K：本轮要找的最小元素的位置序号



(b)选择排序:递增排序示意.

其中三角形代表本轮要找的最小元素应在的位置, 方形代表本轮次至今为止所找到的最小元素所在位置, 三角形左侧为已排好序的元素,三角形右侧的每一元素依次和方形位置元素比较. 实线双向箭头代表两个元素交换. 虚线双向箭头代表两个元素需要比较

内排序基本算法

(3)【冒泡排序法】：基本思想?

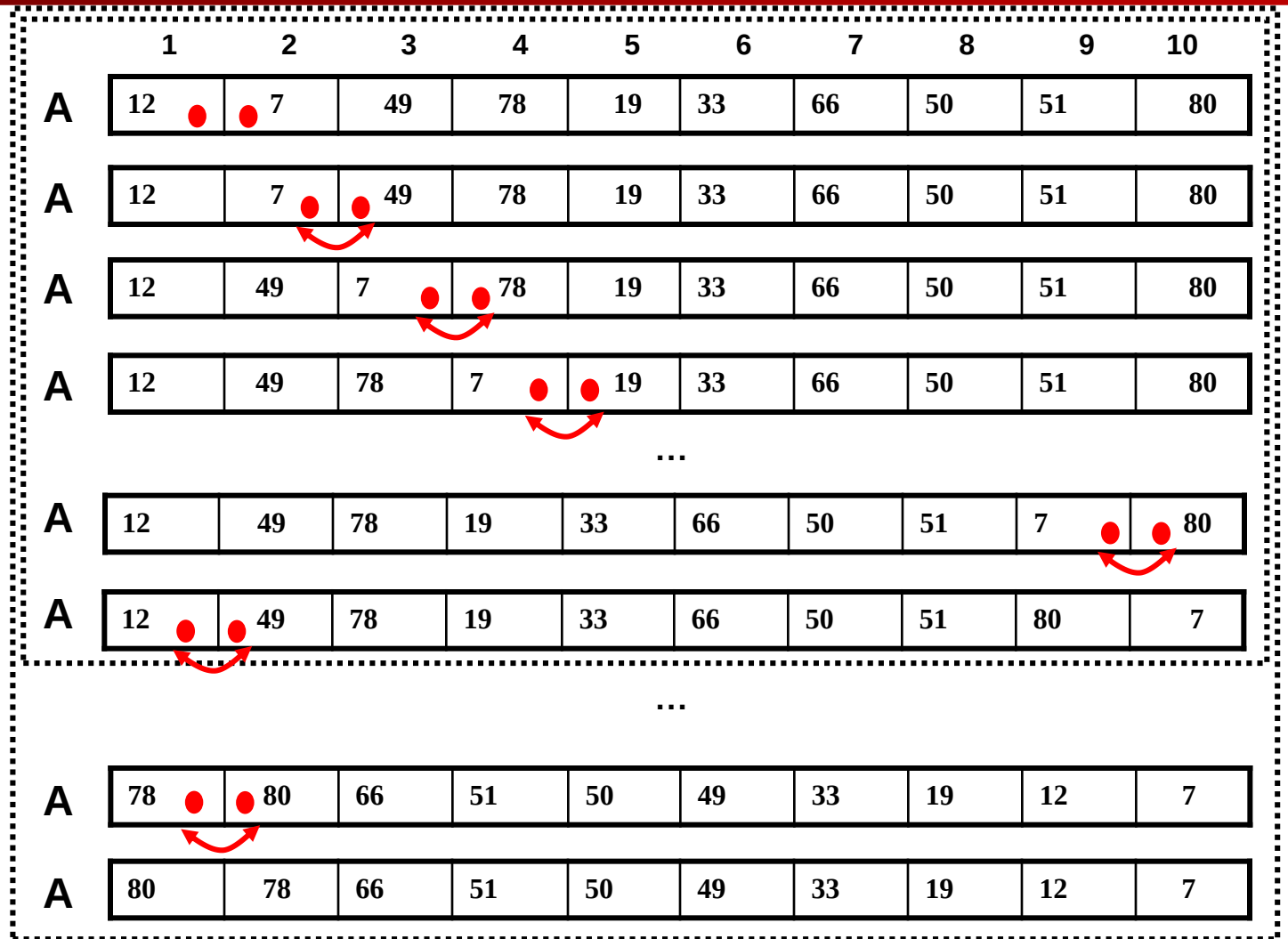
一个轮次一个轮次的处理。

在每一轮次中依次对待排序数组元素中相邻的两个元素进行比较，将大的放前，小的放后--递减排序(或者是将小的放前，大的放后--递增排序)。

当没有交换时，则数据已被排好序。

内排序基本算法

(3)【冒泡排序法】：过程模拟



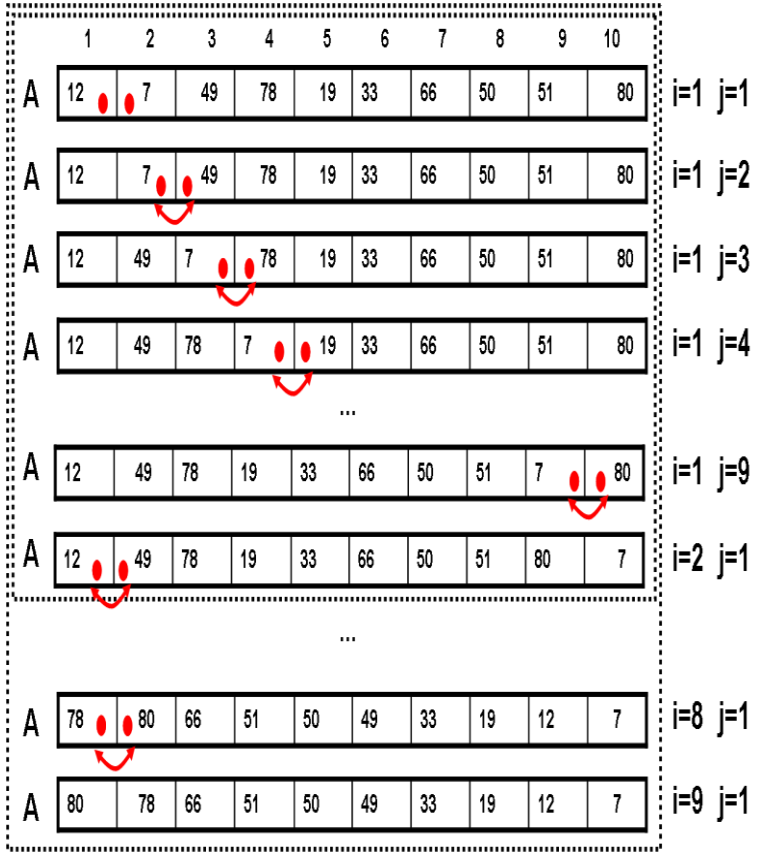
(c)冒泡排序:递减排序示意，其中小圆点代表本轮本次比较的两个元素,双向弧线箭头代表两个元素要相互交换

内排序基本算法

(3)【冒泡排序法】：算法表达

```
BUBBLE-SORT(A)
1.  for i=1 to N-1
2.  {    haschange=false;
3.          for j=1 to N-i
4.          {  if  A[j]>A[j+1]  then
5.                { temp =A[j];
6.                  A[j]=A[j+1];
7.                  A[j+1]=temp;
8.                  haschange=true;
9.                }
10.         }
11.     if (haschange ==false) then break;
12.  }
```

$O(N^2)$



(c)冒泡排序:递减排序示意

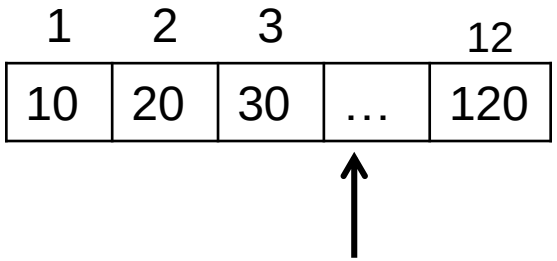
其中小圆点代表本轮本次比较的两个元素,双向弧线箭头代表两个元素要相互交换

三种内排序算法的比较

- 时间复杂度：$O(N^2)$
- 空间复杂度：O(1)
- 执行时间上有差异，比较、交换次数越少的算法越快

算法实例：二分查找

- 二分查找算法：对于一个已经排序好的序列(比如升序)在查找所要查找的元素k时,首先与序列中间的元素进行比较,如果k大于这个元素,就在当前序列的后半部分继续查找,如果k小于这个元素,就在当前序列的前半部分继续查找,直到找到相同的元素,或者所查找的序列范围为空为止。
- 定义数组A[i], i=1,…,n，给定了k值，另外设置变量low, high, mid

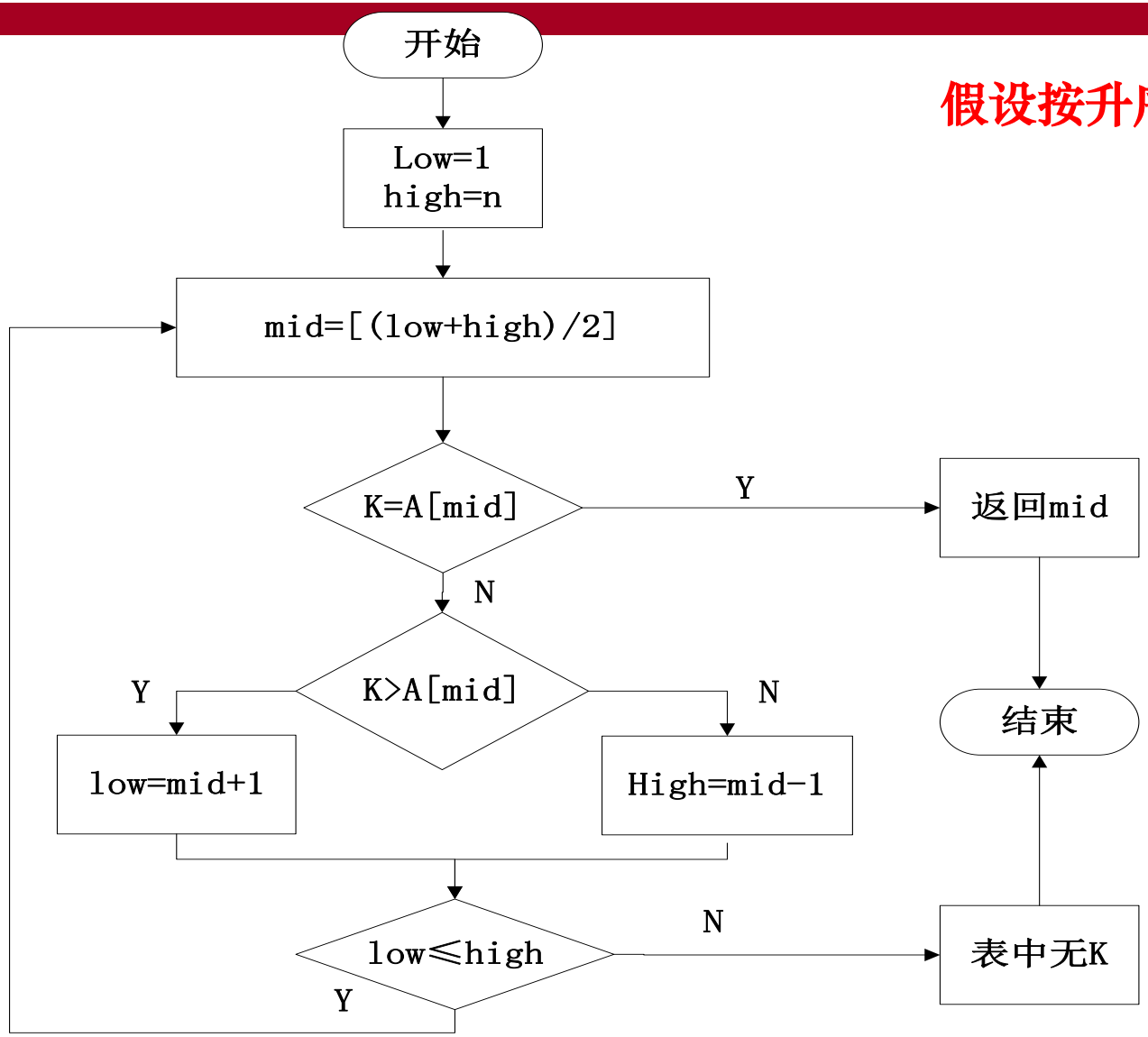


i=n/2=6　A[i]=60

则k=50是在左边or右边?

算法实例：二分查找程序流程图

假设按升序排好序



二分查找实例

1	2	3	4	5	6	7	8	9	10	11
5	13	19	21	37	56	64	75	80	88	92

现在 执行k=21，k=85的过程，给出结论及各变量的值

low	high	mid

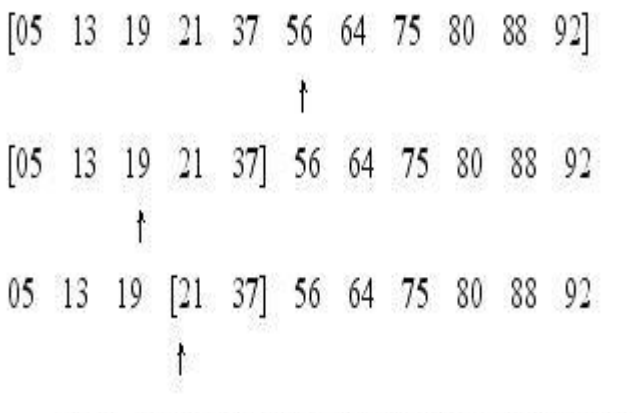
查找k=21过程

low	high	mid
1	11	6
1	5	3
4	5	4

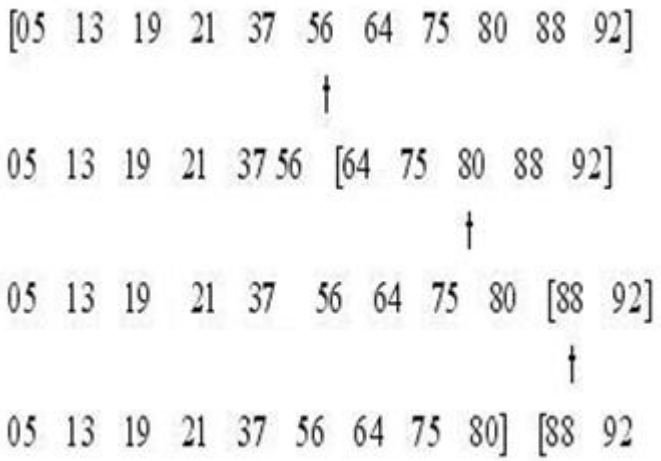
查找k=85过程

low	high	mid
1	11	6
7	11	9
10	11	10
10	9	

二分折半查找过程实例



(a) 查找K=21的过程（3次比较后查找成功）



(b) 查找K=85的过程（3次比较后查找失败）

第10讲-受限资源约束下的算法—排序问题求解示例

一、为什么要研究排序算法

二、内排序基本算法

三、PageRank算法--网页排序

PageRank算法--网页排序问题及思想

(1)网页排序问题：搜索引擎?

4,540,000条检索记录

怎样把最重要的检索记录显示给用户？

1,210,000条检索记录

PageRank算法--网页排序问题及思想

(2)PageRank是什么? 网页又是什么?

问题背景--网页

●PageRank是计算网页重要度的一种方法

网页重要吗?---网页重要度

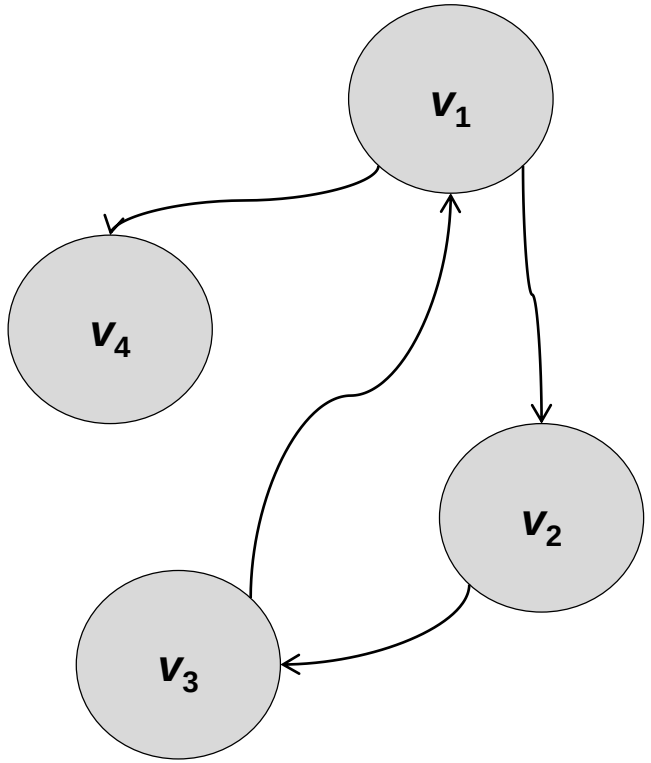
<标记>文本</标记>

<A HREF="http://www.bignet.net/realcorp/products.html">Our Product Information</A>

<A HREF="链接到另一个网页的地址">Our Product Information</A>

PageRank算法—预备知识 （图）

- 图的概念：一个图*G*, 可以表示为一个顶点集 V 和一个边集的 *E* 的组合，即(*V*, *E*)。
- 入度和出度的概念：入度为一个顶点 *v* 被其它顶点指向的次数，出度为一个顶点 *v* 指向别的顶点的次数。

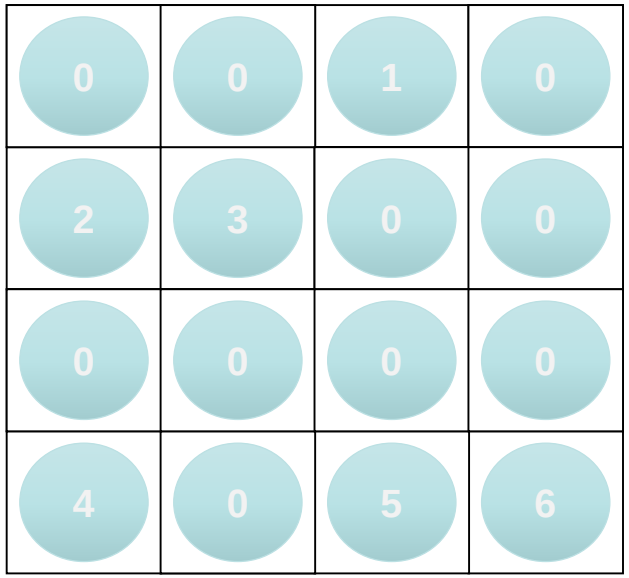


一个有向图*G*，其顶点集 *V* 含四个顶点（v_1，v_2，v_3，v_4），边集 *E* 含四条边（$v_1 \to v_2$ ， $v_1 \to v_4$ ， $v_2 \to v_3$ ， $v_3 \to v_1$ ）。

该图的四个顶点（v_1，v_2，v_3，v_4）的入度分别是1、1、1、1，出度分别为2，1，1，0。

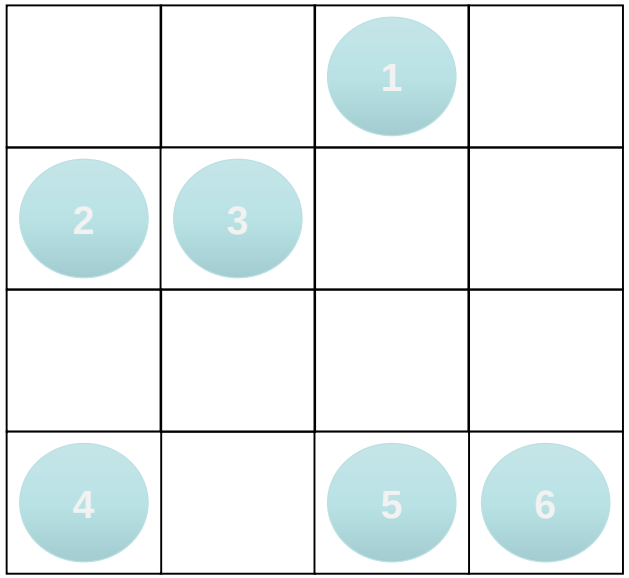
PageRank算法—预备知识（矩阵和稀疏矩阵）

一个矩阵是一个方形的二维数组。如果存储其中的所有值，则称其为“稠密”的。若仅存储其中的非零值，则该矩阵被称为“稀疏矩阵”。



(4x4)

一个“稠密”矩阵

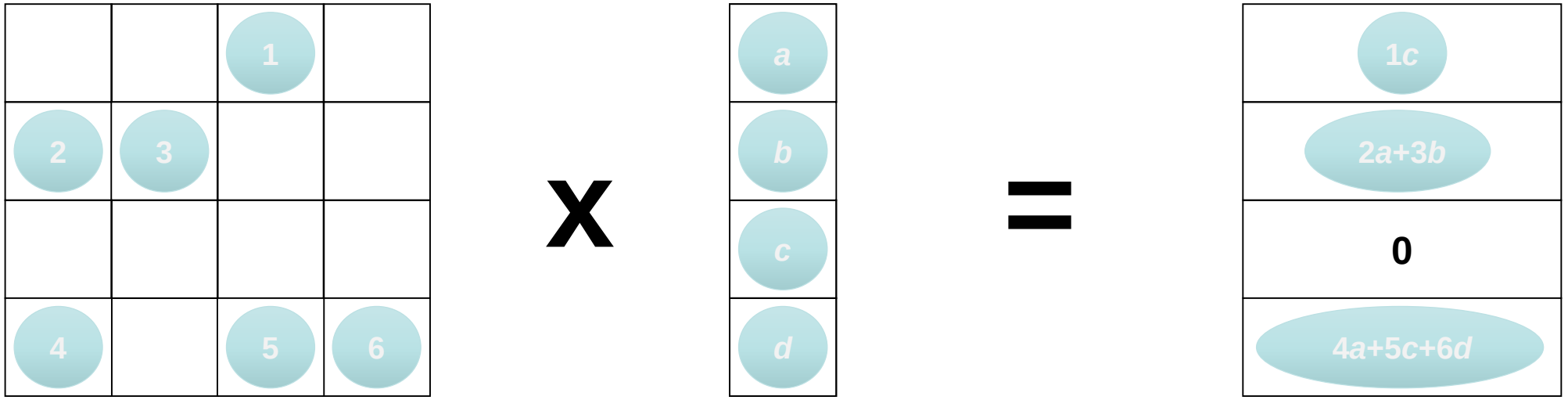


(4x4)

一个“稀疏”矩阵，含6个非零元

PageRank算法—矩阵上的代数操作

使用一个（稠密）向量去乘一个（稀疏）矩阵，获得一个（稠密）向量。

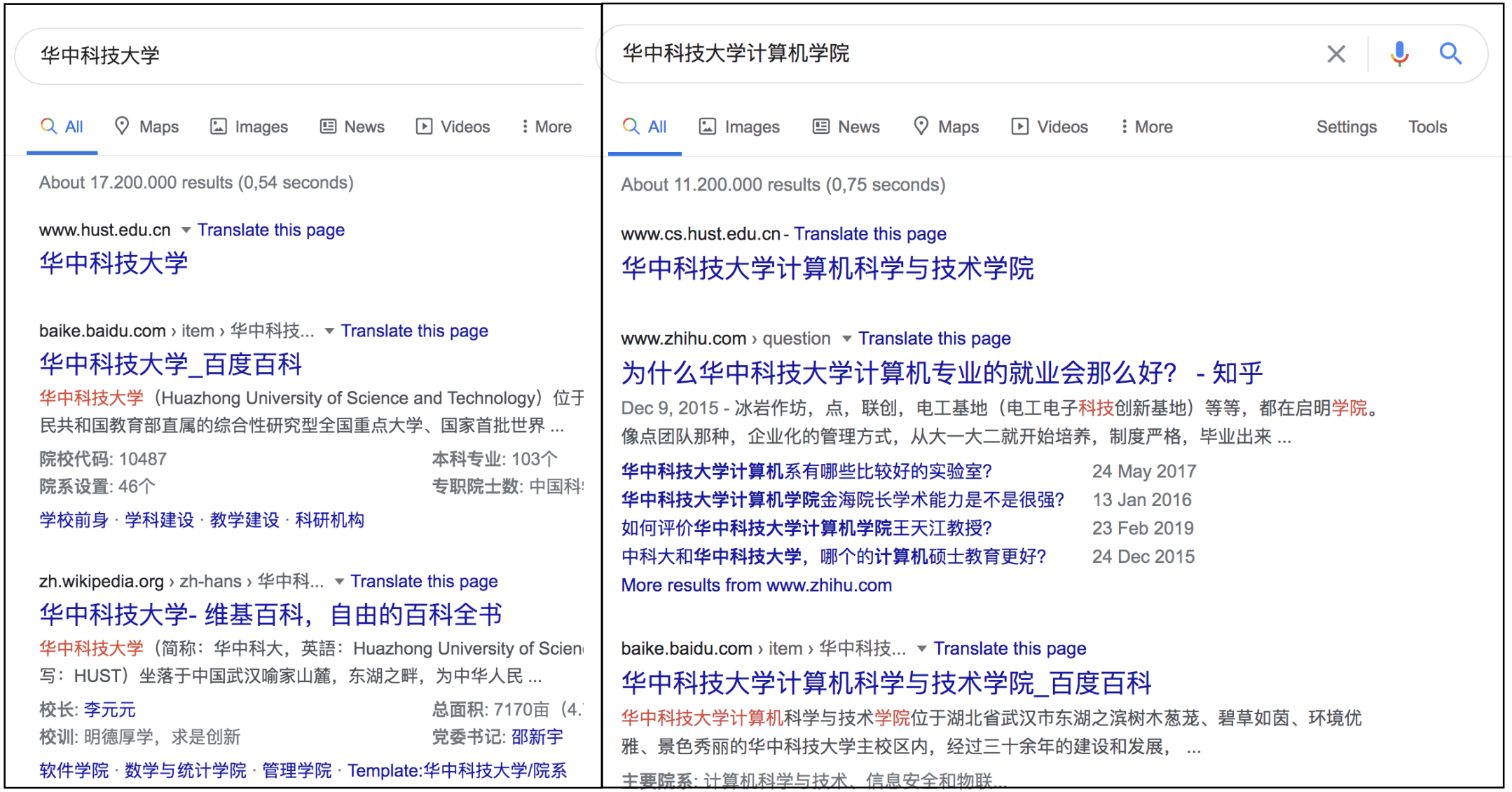


A
(4x4)
稀疏矩阵

x
(4x1)
稠密向量

y
(4x1)
稠密向量

PageRank算法--网页排序问题及思想



华中科技大学

All Maps Images News Videos More

About 17.200.000 results (0,54 seconds)

www.hust.edu.cn ▾ Translate this page

华中科技大学

baike.baidu.com › item › 华中科技... ▾ Translate this page

华中科技大学_百度百科

华中科技大学（Huazhong University of Science and Technology）位于民共和国教育部直属的综合性研究型全国重点大学、国家首批世界 ...

院校代码: 10487　　本科专业: 103个

院系设置: 46个　　专职院士数: 中国科

学校前身 · 学科建设 · 教学建设 · 科研机构

zh.wikipedia.org › zh-hans › 华中科... ▾ Translate this page

华中科技大学- 维基百科，自由的百科全书

华中科技大学（简称：华中科大，英語：Huazhong University of Scien

写：HUST）坐落于中国武汉喻家山麓，东湖之畔，为中华人民 ...

校长: 李元元　　总面积: 7170亩（4.

校训: 明德厚学，求是创新　　党委书记: 邵新宇

软件学院 · 数学与统计学院 · 管理学院 · Template:华中科技大学/院系

华中科技大学计算机学院

All Images News Maps Videos More Settings Tools

About 11.200.000 results (0,75 seconds)

www.cs.hust.edu.cn - Translate this page

华中科技大学计算机科学与技术学院

www.zhihu.com › question ▾ Translate this page

为什么华中科技大学计算机专业的就业会那么好？ - 知乎

Dec 9, 2015 - 冰岩作坊，点，联创，电工基地（电工电子科技创新基地）等等，都在启明学院。像点团队那种，企业化的管理方式，从大一大二就开始培养，制度严格，毕业出来 ...

华中科技大学计算机系有哪些比较好的实验室? 24 May 2017

华中科技大学计算机学院金海院长学术能力是不是很强? 13 Jan 2016

如何评价华中科技大学计算机学院王天江教授? 23 Feb 2019

中科大和华中科技大学，哪个的计算机硕士教育更好? 24 Dec 2015

More results from www.zhihu.com

baike.baidu.com › item › 华中科技... ▾ Translate this page

华中科技大学计算机科学与技术学院_百度百科

华中科技大学计算机科学与技术学院位于湖北省武汉市东湖之滨树木葱茏、碧草如茵、环境优雅、景色秀丽的华中科技大学主校区内，经过三十余年的建设和发展， ...

主要院系: 计算机科学与技术、信息安全和物联...

PageRank算法--网页排序问题及思想

Google 创始人之一Larry Page在1998年提出

The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

Lawrence Page拉里佩奇

Sergey Brin谢尔盖布林

PageRank算法--网页排序问题及思想

(2)Google的网页排序

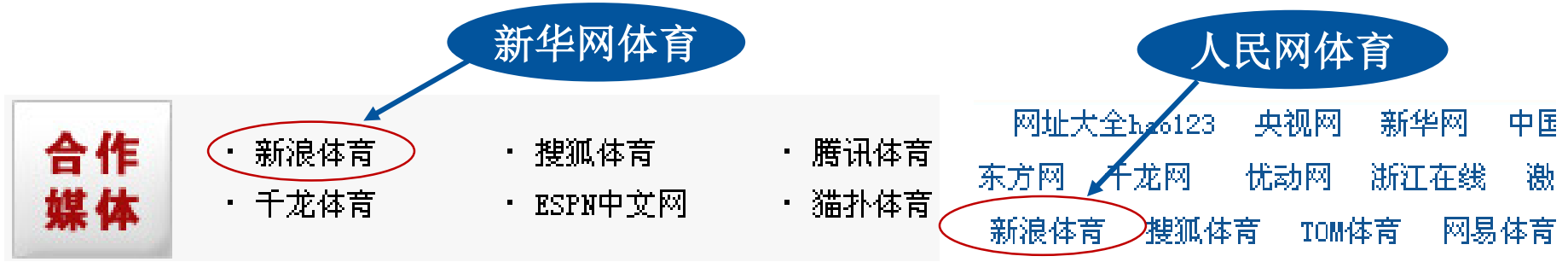
- 在Google中搜索“华中科技大学”
- 搜索引擎工作的简要过程如下：
  - 针对查询词“华中科技大学”进行分词=》华中、科技、大学
  - 根据建立的倒排索引，将同时包含华中、科技和大学的文档返回，并根据相关性进行排序
  - 但是会有一些垃圾网页，虽然包含大量的查询词，但却并非满足用户需求的文档
  - 页面本身的重要性在网页排序中也起着重要作用

基于内容的相关性

PageRank算法--网页排序问题及思想

(2)PageRank是什么? 网页又是什么?

- 如何度量网页本身的重要性呢?
  - 比如，新华网体育在其首页中对新浪体育做了链接，人民网体育同样在其首页中对新浪体育做了链接

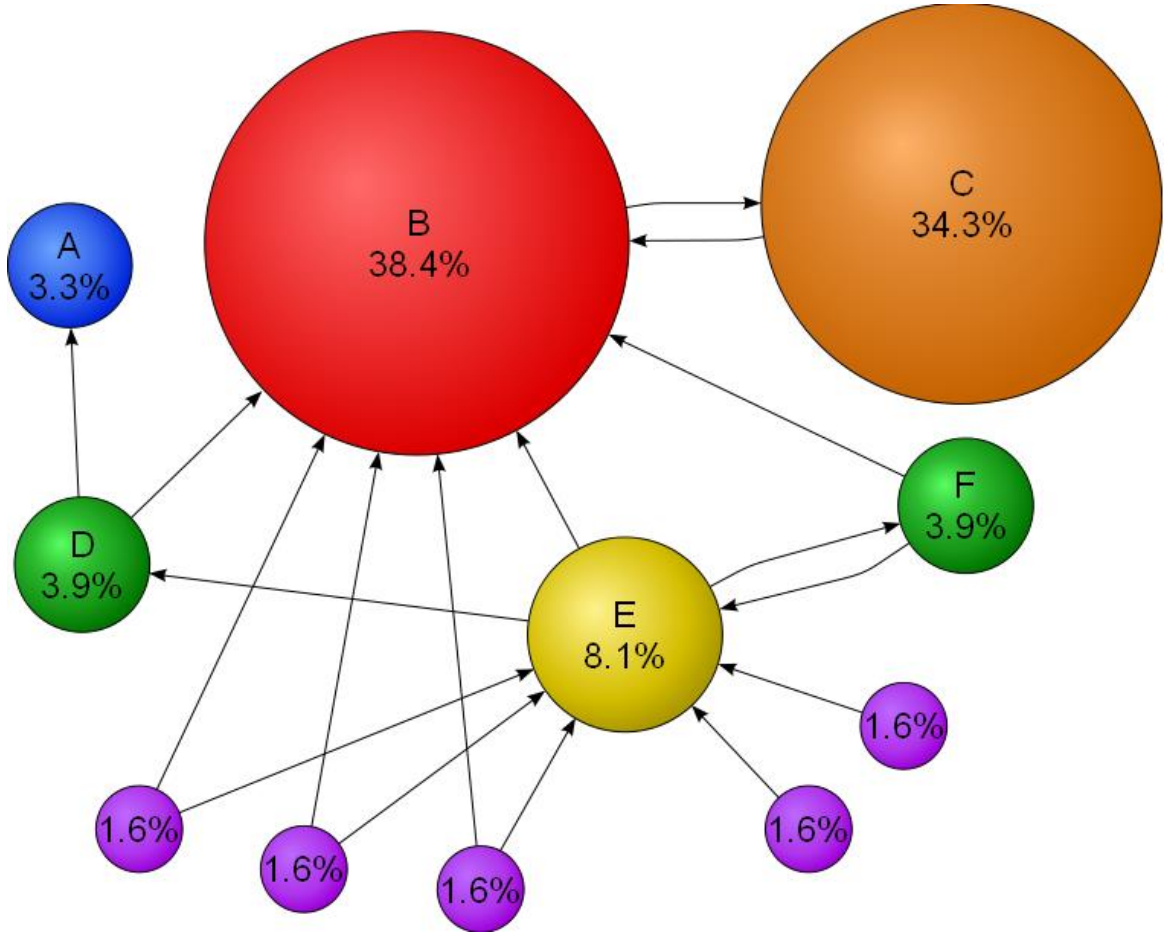


- 可见，新浪体育被链接的次数较多；同时，人民网体育和新华网体育也都是比较“重要”的网页，因此新浪体育也应该是比较“重要”的网页。

PageRank算法--网页排序问题及思想

(2)PageRank是什么? 网页又是什么?

- 一个更加形象的图

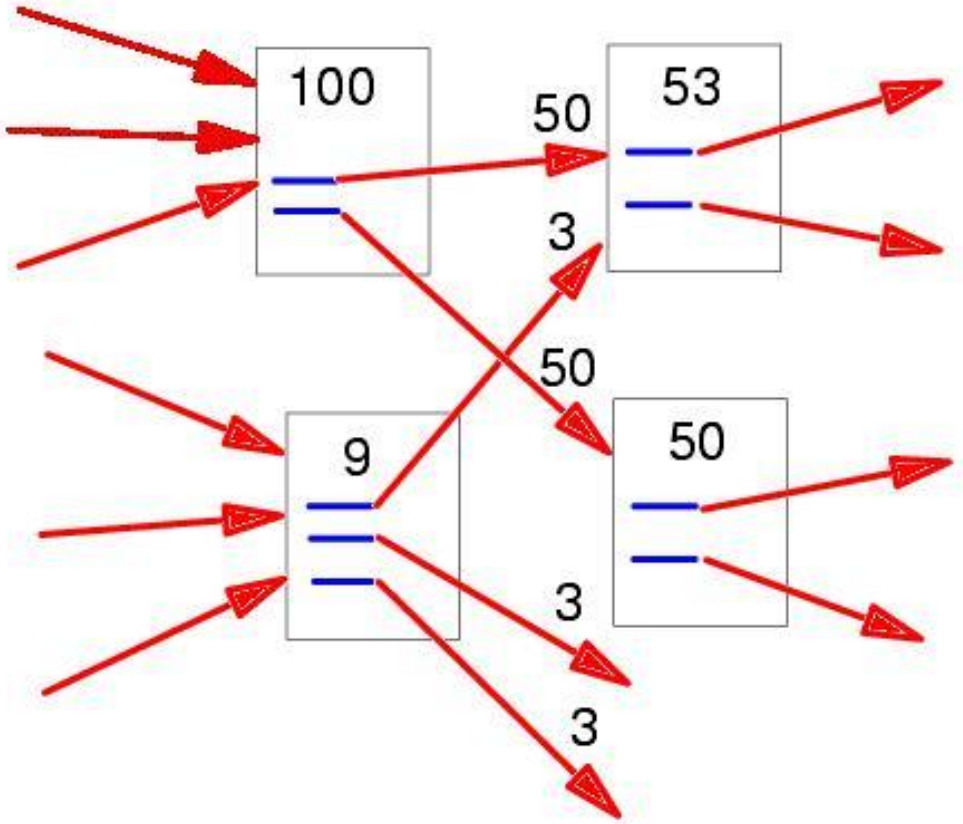


链向网页E的链接远远多于链向网页C的链接，但是网页C的重要性却大于网页E。这是因为因为网页C被网页B所链接，而网页B有很高的重要性。

PageRank算法--网页排序问题及思想

(2)PageRank是什么? 网页又是什么?

•**PageRank** 是基于「从许多优质的网页链接过来的网页，必定还是优质网页」的回归关系，来判定所有网页的重要性。



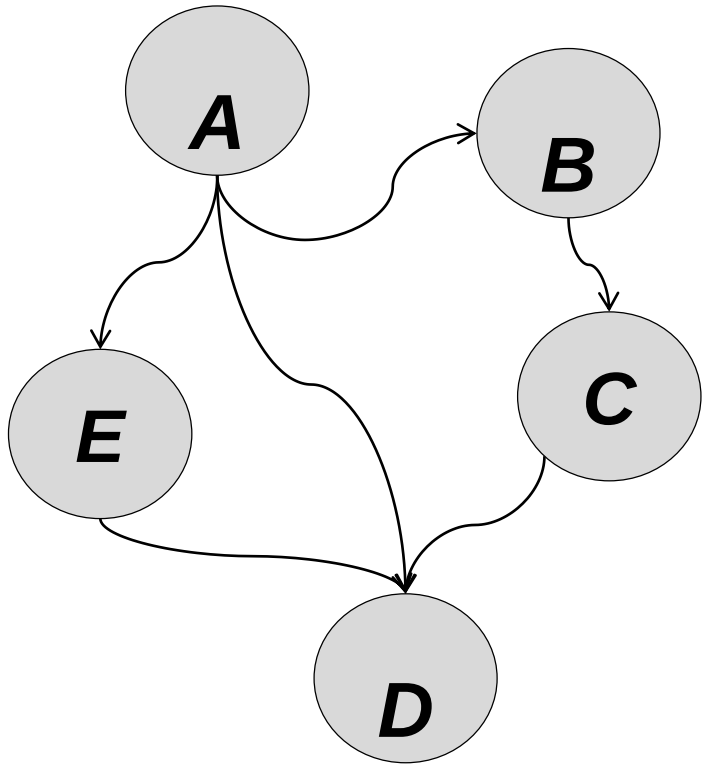
•链入链接数 (单纯的意义上的受欢迎度指标)

•链入链接是否来自推荐度高的页面 (有根据的受欢迎指标)

•链入链接源页面的链接数 (被选中的几率指标)

PageRank算法--网页排序问题及思想

(2)PageRank算法的简单实现



以左图中的顶点页面*D*为例，其页面PageRank值，标记为*PR*（*D*），可以通过以下公式计算：

$PR(D) = PR(A)/3 + PR(C)/1 + PR(E)/1$ ，

意味着*PR*(*A*)的1/3和全部*PR*(*C*)及*PR*(*E*)贡献到*PR*(*D*)。

归纳一下，顶点*u*的PageRank值为

$$PR(u) = \sum_{v \epsilon P_u} \frac{PR(v)}{OD(v)}$$

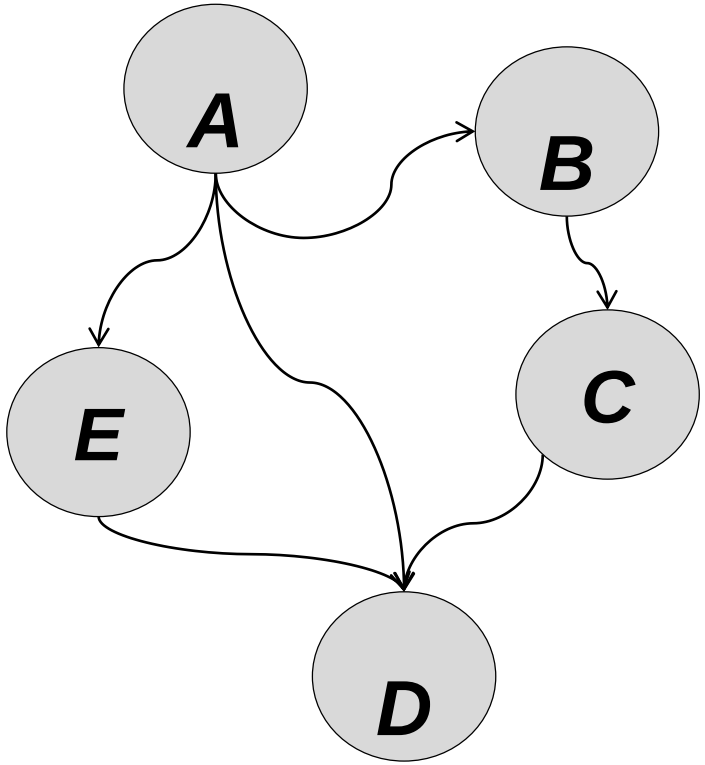
其中*OD*(*v*)是顶点*v*的出度，P_u 是链接到顶点*u*所有顶点的集合。

Question: 为什么是*PR*(*A*)的1/3？

Answer: 当一个用户在浏览页面*A*（链接到三个页面*B*、*D*、*E*）时，点击链接到*D*的概率是1/3。

PageRank算法--网页排序问题及思想

(2)PageRank算法的简单实现

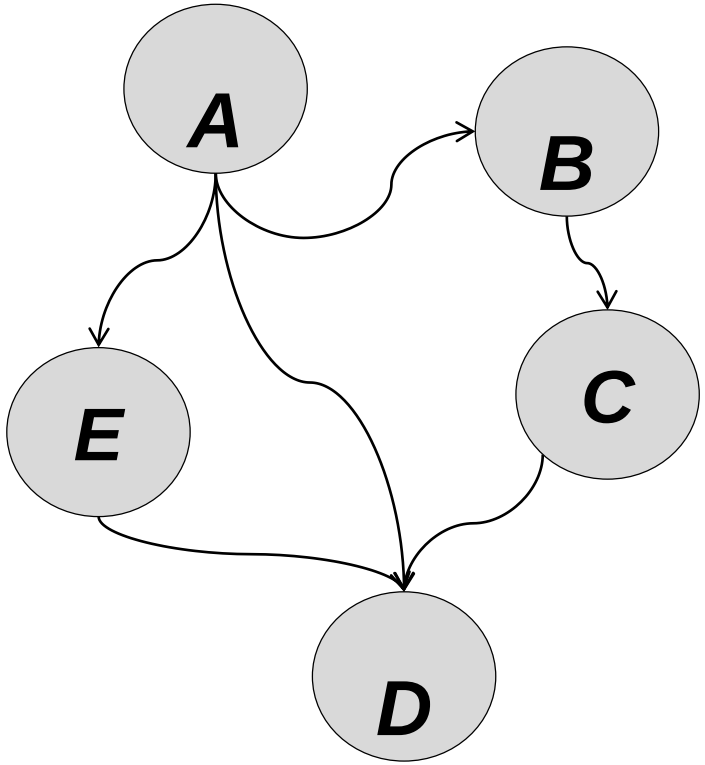


PageRank算法是一种“迭代”方法，即求解开始时给每一个待求解分量（一个页面的PageRank值）一个任意的初值，再执行算法若干次（即若干个“迭代步”），直到最新得出的分量“足够”接近上一次的分量（即算法收敛）。

在我们的例子中，设定*PR*(*A*), ... ,*PR*(*E*)均为 0.2 (即1/*N*，*N*为5), 然后在每一个迭代步计算直至收敛。

PageRank算法--网页排序问题及思想

(2)PageRank算法面临的两个问题



某些页面没有出链 （如页面D），会发生什么?

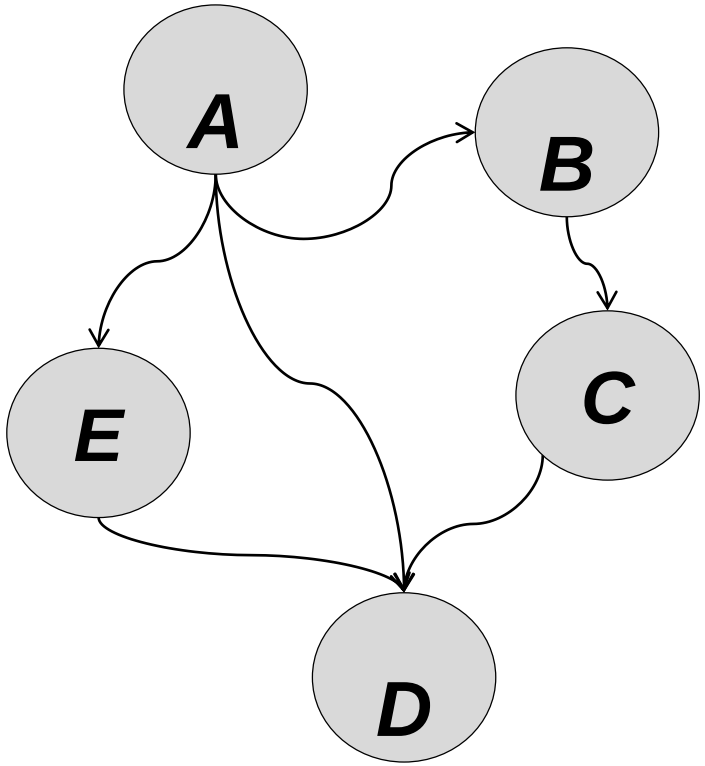
Rank Leak：某些页面没有出链，像黑洞一样，吸收其他网页的影响力而不释放，最终会导致其他网页的PR值为0.

某些页面没有入链 （如页面A），会发生什么?

Rank Sink：某些页面没有入链，计算的过程迭代下来，会导致这个网页的PR值为0。

PageRank算法--网页排序问题及思想

(2)PageRank算法的随机浏览模型



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在哪个页面，都有概率访问到其他的任意页面。

用户以一定概率d继续按照现有超链接形式访问，以概率1-d访问随机一个页面。

进一步，我们改进pagerank公式为

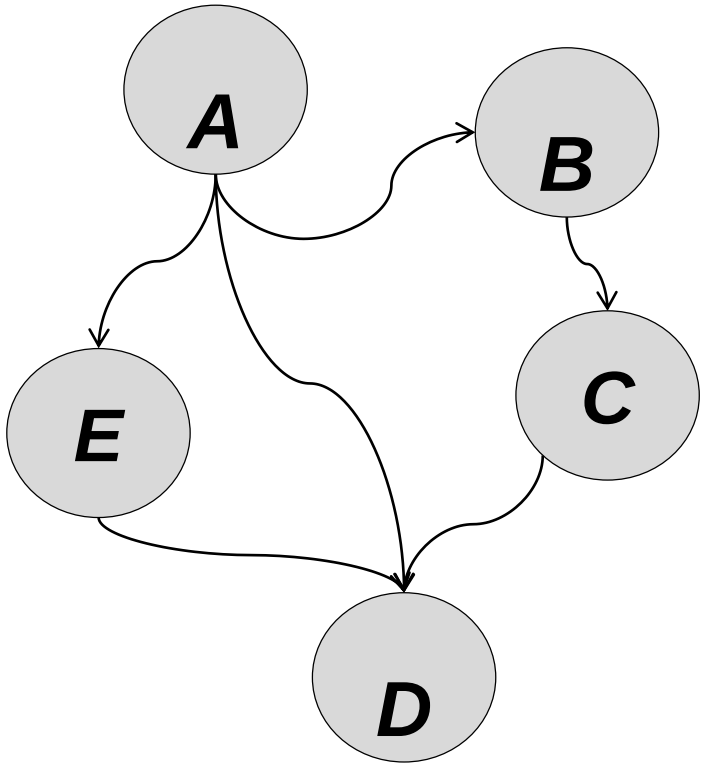
$$PR(u) = \frac{1-d}{N} + d\sum_{v \in P_u} \frac{PR(v)}{OD(v)}$$

d也被成为阻尼因子，根据工程经验，一般设为0.85；

*N*为页面的总数。

PageRank算法--网页排序问题及思想

(2)PageRank算法的简单实现

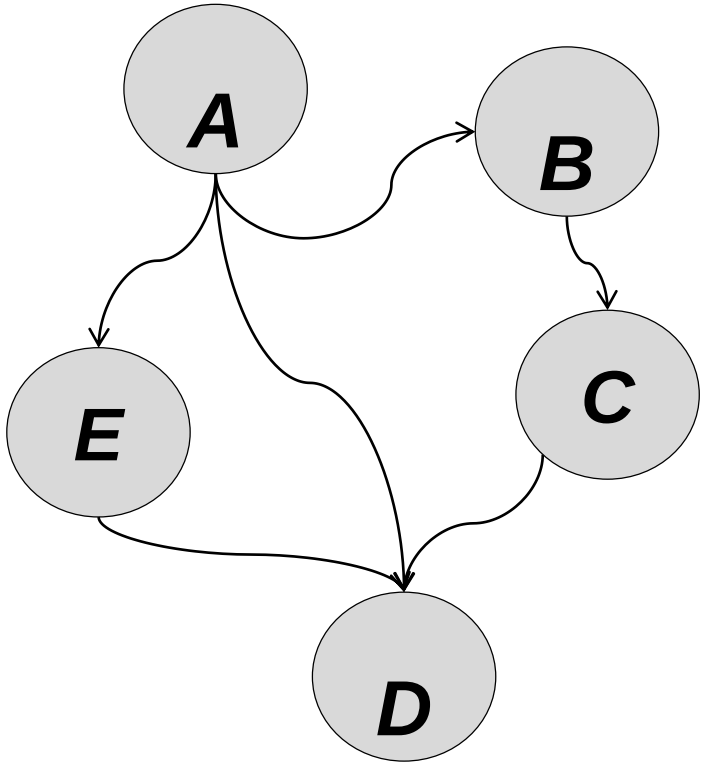


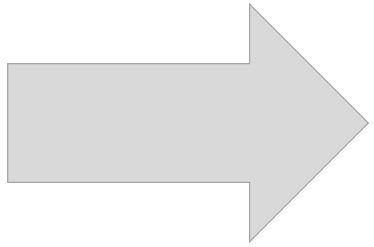
在我们的例子中，设定*PR*(*A*), ... ,*PR*(*E*)均为 0.2 (即1/*N*，*N*为5), 然后在每一个迭代步计算以下公式直至收敛。

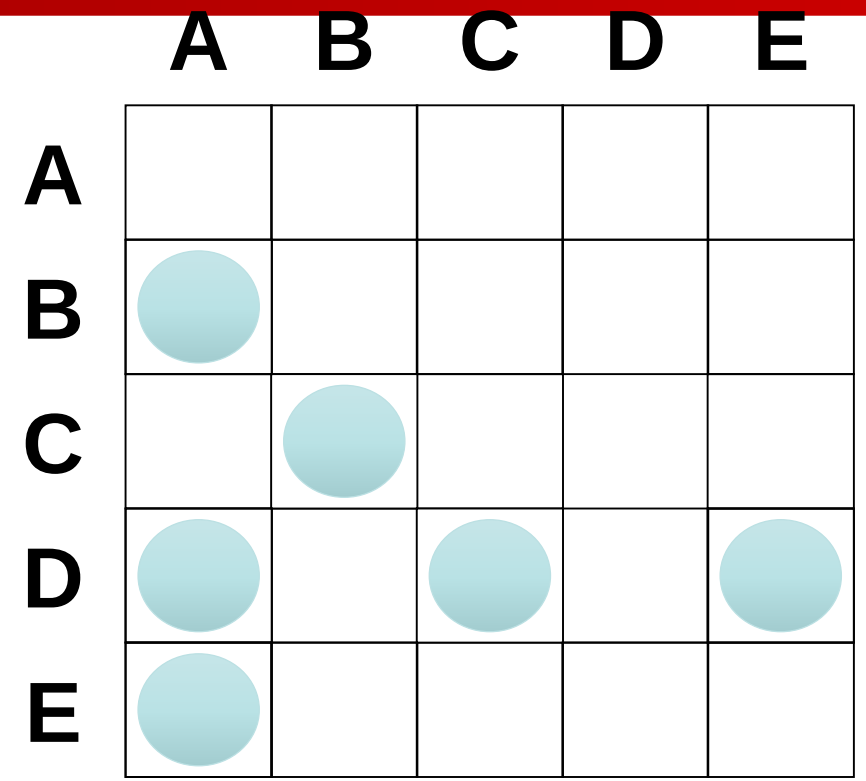
$$PR(A) = 0.15 / 5$$
$$PR(B) = 0.15 / 5 + 0.85(PR(A) / 3)$$
$$PR(C) = 0.15 / 5 + 0.85(PR(B) / 1)$$
$$PR(D) = 0.15 / 5 + 0.85(PR(A) / 3 + PR(C) / 1 + PR(E) / 1)$$
$$PR(E) = 0.15 / 5 + 0.85(PR(A) / 3)$$

PageRank算法--网页排序问题及思想

(2)PageRank的矩阵化





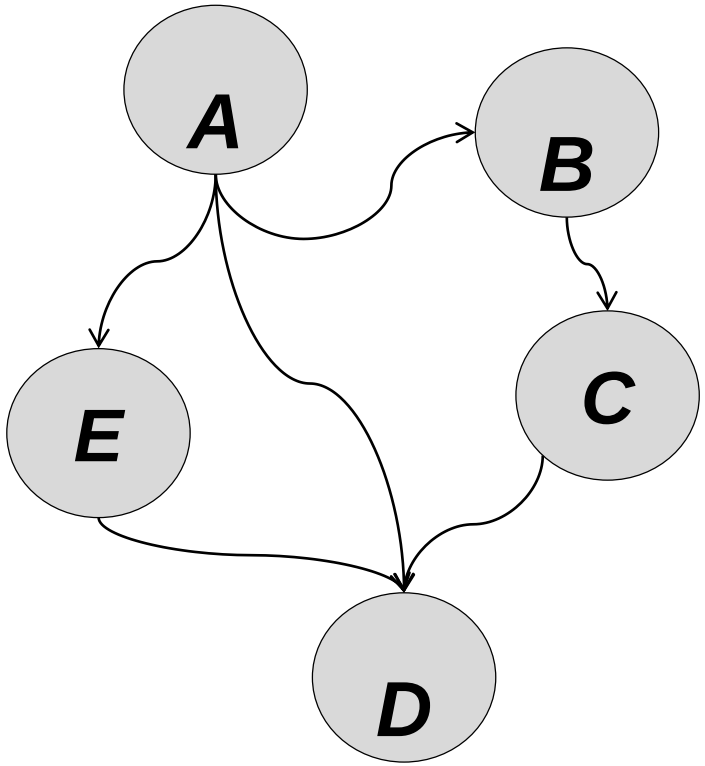


M

将“图”转化为“稀疏矩阵”，其中列中的非零元为图相应顶点的中的链接状态。

PageRank算法--网页排序问题及思想

(2)PageRank的矩阵化



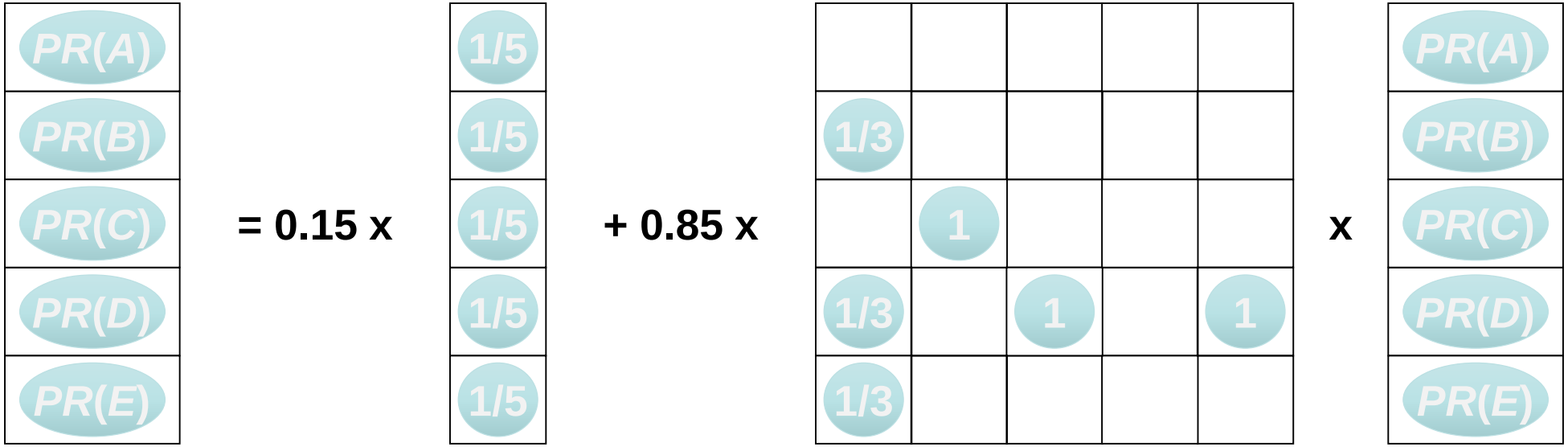
	A	B	C	D	E
A					
B	1/3				
C		1			
D	1/3		1		1
E	1/3				

M

该稀疏矩阵中的非零元值对应为相应顶点的出度信息。

PageRank算法--网页排序问题及思想

（2）使用线性代数实现PageRank算法



以上线性代数方法对应的即是之前提及的PageRank基础算法：

$$PR(A) = 0.15 / 5$$

$$PR(B) = 0.15 / 5 + 0.85(PR(A) / 3)$$

$$PR(C) = 0.15 / 5 + 0.85(PR(B) / 1)$$

$$PR(D) = 0.15 / 5 + 0.85(PR(A) / 3 + PR(C) / 1 + PR(E) / 1)$$

$$PR(E) = 0.15 / 5 + 0.85(PR(A) / 3)$$