第2章 万维网网页信息的表达及解析

2.1 万维网架构及网页表达

2.2 网页信息抽取

2.3 搜索引擎与分析系统

2.4 搜索引擎索引系统

2.5 搜索引擎查询系统5

《走进搜索引擎》 潘雪峰 花贵春 梁斌编著
电子工业出版社 2011年5月第2版

第2章 万维网网页信息的表达及解析

2.2 网页信息抽取（2章）

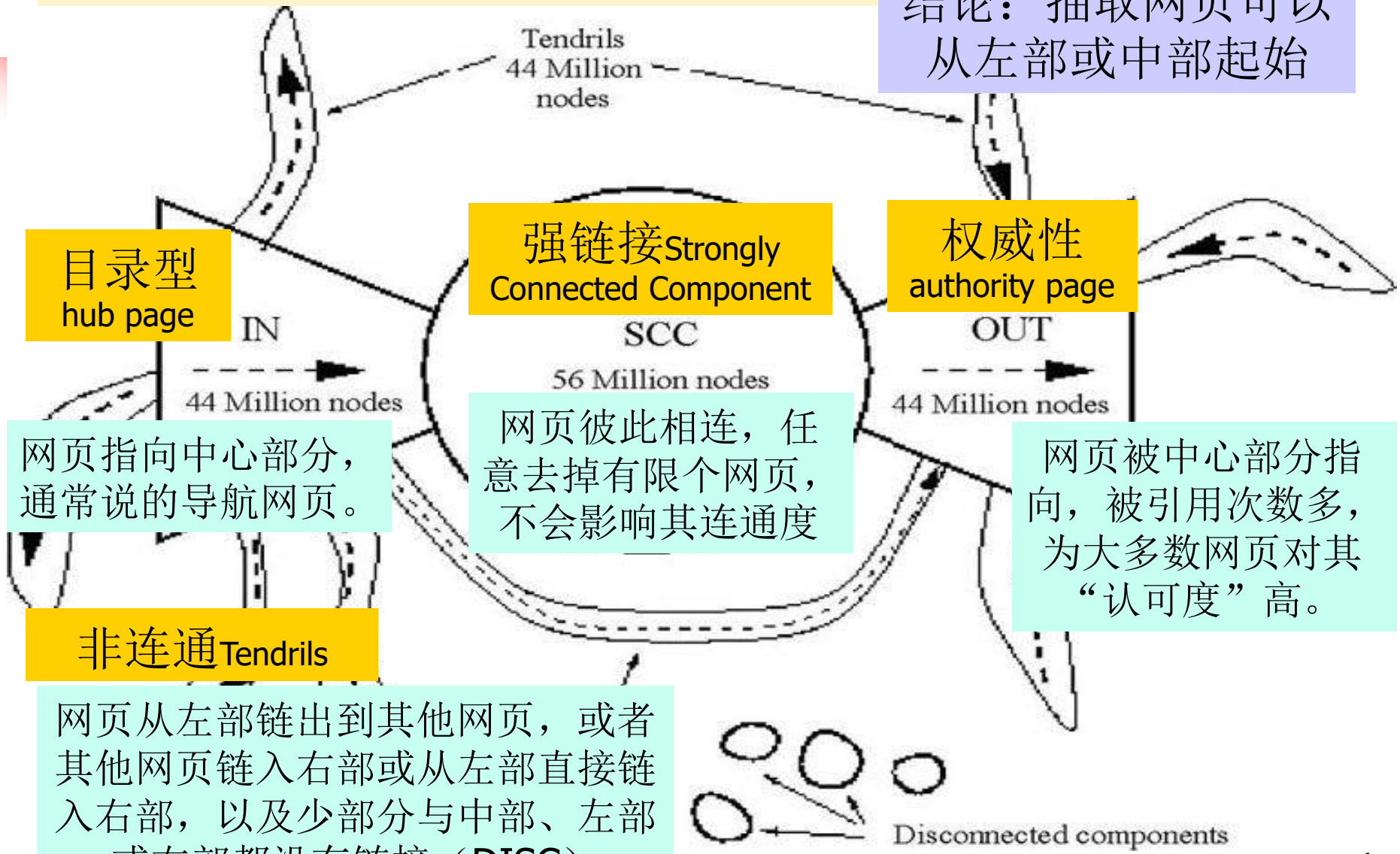
- 2.2.1 万维网结构特点2.2
- 2.2.2 网页抓取原理2.4
- 2.2.3 网页查重3.3
- 2.2.4 内容扩展：Jaccard系数

2.2.1 万维网结构特点2.2

- 万维网的静态结构：
  - 是一个相互连通的连通图：
    - 网页为结点
    - 链接（link）为边
      - “反向链接”（back link）：任意一个网页可能被其他网页链接
      - “正向链接”（简称“链接”）：可能链接到其他网页
- 网页的3大特征
  - 挥发性（volatile）：网页诞生后到网页的消亡（挥发性）
  - 半结构性（semi-structured）：HTML语言描述的网页是一种半结构化的数据。
  - 隐蔽性（hidden）：除了静态网页以外，还有很多隐藏的动态网页，例如需要登录才能看到的某些动态网页。

- 通过实验得出万维网具有蝴蝶结型（bow tie）结构
  - 网页4种类型，各约占1/4

结论：抽取网页可以从左部或中部起始



目录型
hub page

强链接Strongly Connected Component

权威性
authority page

网页指向中心部分，通常说的导航网页。

网页彼此相连，任意去掉有限个网页，不会影响其连通度

网页被中心部分指向，被引用次数多，为大多数网页对其“认可度”高。

非连通Tendrils

网页从左部链出到其他网页，或者其他网页链入右部或从左部直接链入右部，以及少部分与中部、左部或右部都没有链接（DISC）

第2章 万维网网页信息的表达及解析

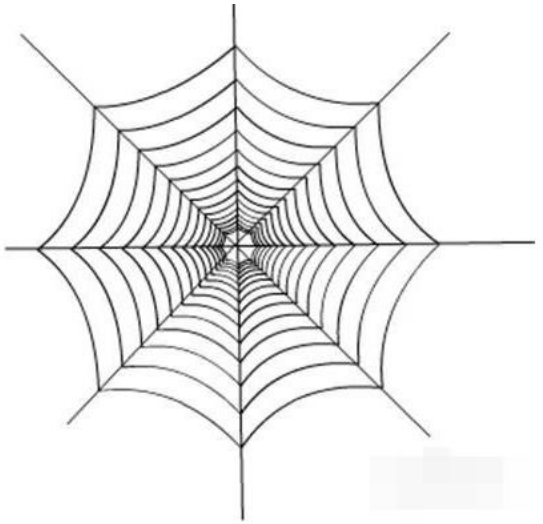
2.2 网页信息抽取（2章）

2.2.1 万维网结构特点2.2

2.2.2 网页抓取原理2.4

2.2.3 网页查重3.3

2.2.4 内容扩展：Jaccard系数



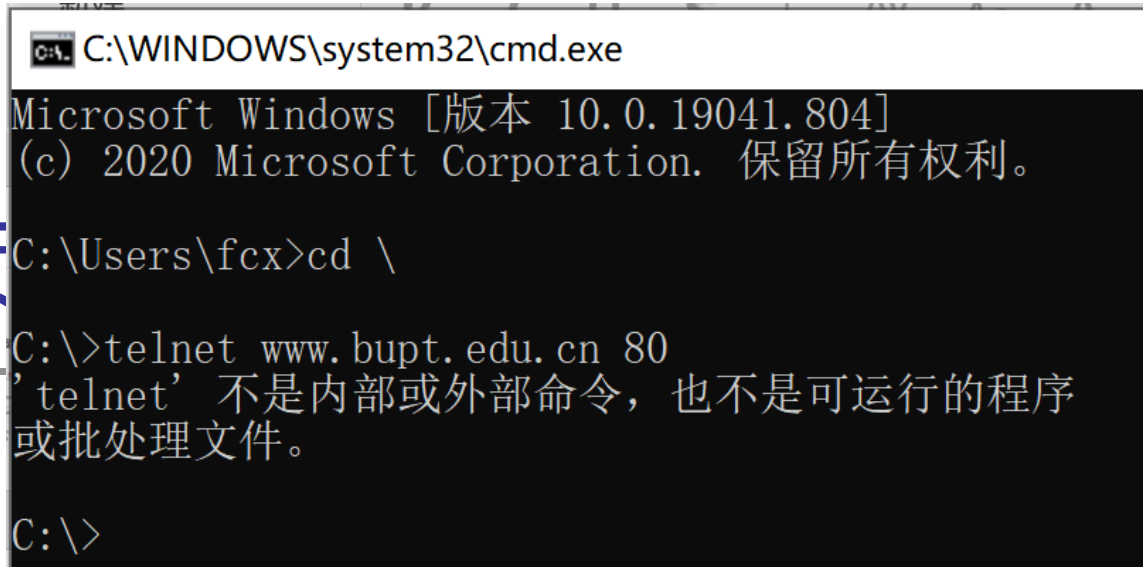
2.2.2 网页抓取原理

- 网页信息抽取：
  - 自动抓取万维网网页信息
  - 机器人(爬虫Crawler或者“蜘蛛”)——能够检索几乎全部的万维网网页
- 世界上第1个网络爬虫由麻省理工学院的学生马休·格雷（Matthew Gray）在1993年写成，并命为“万维网漫游者Wanderer”。
- 现代搜索引擎的思路源于Wanderer：
  - 1994年7月，Michael Mauldin将John Leavitt的蜘蛛程序接入到其索引程序中，创建了当时著名的搜索引擎Lycos（http://www.lycos.com/）。
  - 其后无数的搜索引擎促使爬虫越写越复杂，并逐渐向多策略、负载均衡及大规模增量抓取等方向发展。

2.2.2 网页抓取原理

- 1.网页抓取几个概念
  - 种子站点
    - 爬虫开始抓取的起点，如SCC、IN
  - URL（统一资源定位器）
    - 用在万维网和其他万维网资源中的一种编址系统
  - Backlinks（反向链接）
    - 除网页自身之外指向自身链接的集合
    - Backlinks的数目是衡量网页受“欢迎”程度的重要度量方式

2.2.2 网页



- 2.网页获取关键问题
- 通过操作系统命令下载一个网页（手动）
  - 使用Windows操作系统的用户：

  （1）打开Windows命令行窗口。

  （2）输入telnet www.bupt.edu.cn 80

  （3）输入GET /index.html（注意GET要全大写，其后要有空格）
  - 在桌面上看到北京邮电大学大学网站首页的网页源代码
  - （需要本机权限及对方允许权限，如果Windows没有自动执行telnet权限，则不能执行）

2.2.2 网页抓取原理

获取一个网页（片段）

```
        <!-- IE 8 及以下跳转至浏览器升级页n -->
                <!--[if lt IE 9]>
                        <script>
                                window.location.href = "/statics/ieup.html"
                        </script>
                <![endif]-->
<title>欢迎访问信息服务门户</title>

        <script type="text/javascript" src="script/top-simple.js?v=2.51"></script>



<script type="text/javascript">
   jQuery.noConflict();
```

2.2.2 网页抓取原理

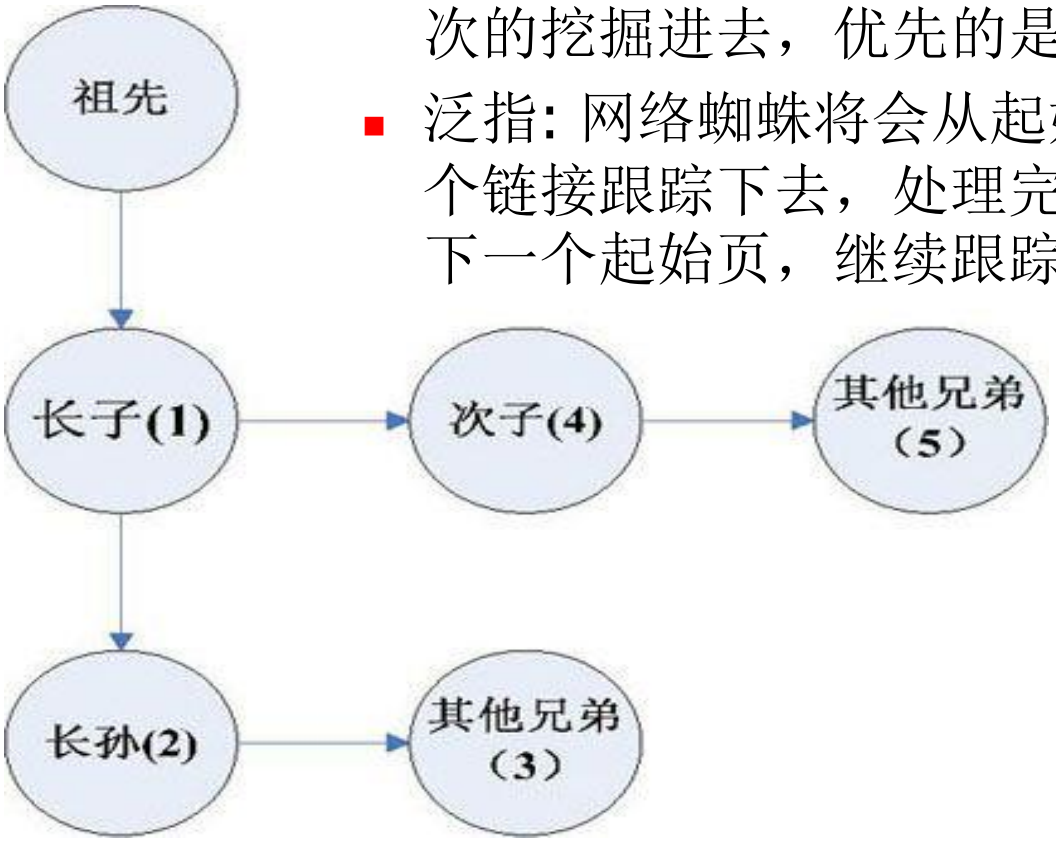
遍历原则：
深度优先策略
宽度优先策略

- 获取所有网页相关工作：
  - 遍历万维网：
    - 要下载整个万维网，需要遍历万维网
  - 抓取“顺序”策略：
    - 万维网的蝴蝶结型结构，非线性的网页组织结构
    - 抓取"顺序"的策略必须保证尽可能抓取所有网页
      - 一般来说，爬虫选择蝴蝶结左部的网页。即目录型网页作为种子站点（抓取出发点），典型的如sina.com和sohu.com这样的门户网站的主页。
  - 提取链接:
    - 每完成一次抓取网页之后，提取指向其他网页的URL，它们指引爬虫更加深入地抓取其他网页
  - 继续抓取

2.2.2 网页抓取原理

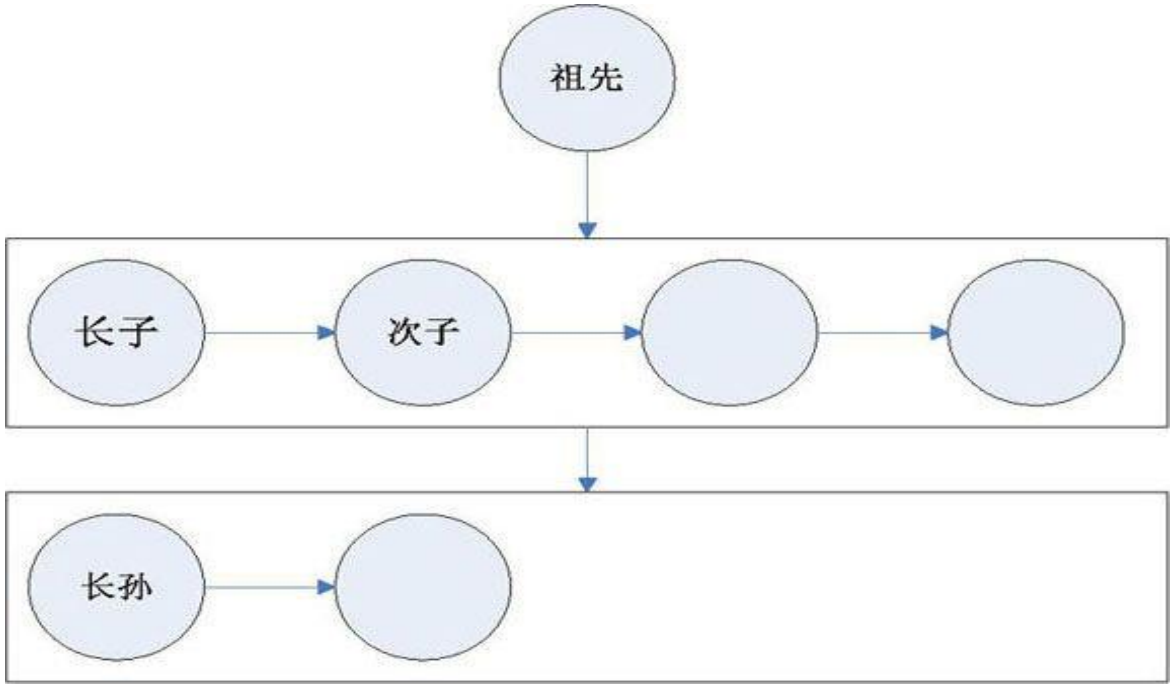
3.深度优先策略（Depth-First Traversal）

- 网络蜘蛛尽量的在抓取网页时，往网页更深层次的挖掘进去，优先的是深度!
- 泛指: 网络蜘蛛将会从起始页开始，一个链接一个链接跟踪下去，处理完这条线路之后再转入下一个起始页，继续跟踪链接



4.宽度优先策略(广度优先/层次优先)(Breadth-First Traversal)

- 先搜索完一个Web 页面中所有的超级链接，然后再继续搜索下一层, 直到底层为止。
  - 例如，一个HTML 文件中有三个超链,选择其中之一并处理相应的HTML文件，然后不再选择第二个HTML文件中的任何超链, 而是返回并选择第二个超链，处理相应的HTML文件，再返回，选择第三个超链并处理相应的HTML文件。



一层上的所有超链都已被选择过，就可以开始在刚才处理过的HTML 文件中搜索其余的超链

2.2.2 网页抓取原理

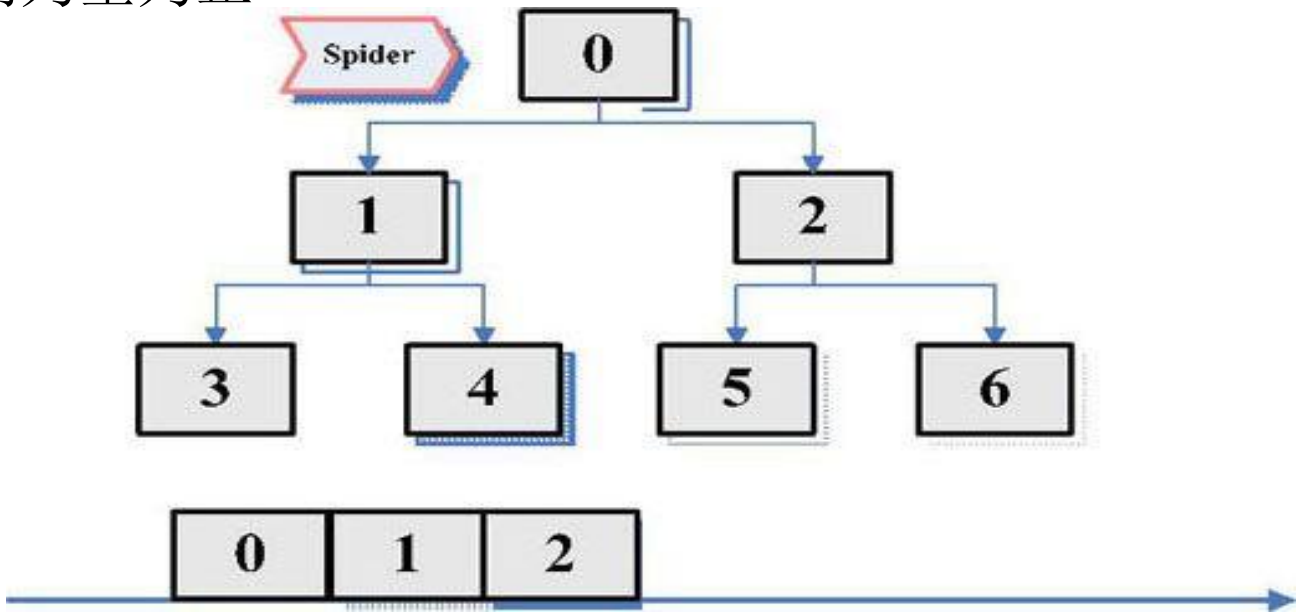
网页抓取的顺序策略上选择宽度优先

- 宽度优先策略优点(针对网页抽取）：
  - 保证了对浅层的首先处理
    - 重要的网页往往离种子站点的距离较近，深度加深，网页的重要性越来越低
  - 当遇到一个无穷尽的深层分支时，不会导致陷进WWW 中的深层文档中出现出不来的情况发生。
  - 能在两个HTML文件之间找到最短路径。
  - 中文万维网直径的长度只有17-19，到达某一个网页的路径通常很多，总会存在一条很短的路径到达。
  - 有利于多爬虫合作抓取，抓取的封闭性较强。

2.2.2 网页抓取原理

5.宽度优先抓取网页：

- 设一个队列做数据结构，队列表达工作负载
- 只要队列中存在没有完成的抓取任务，就需要提取队头位置的网页继续抓取。直到完成全部抓取任务，工作负载队列为空为止

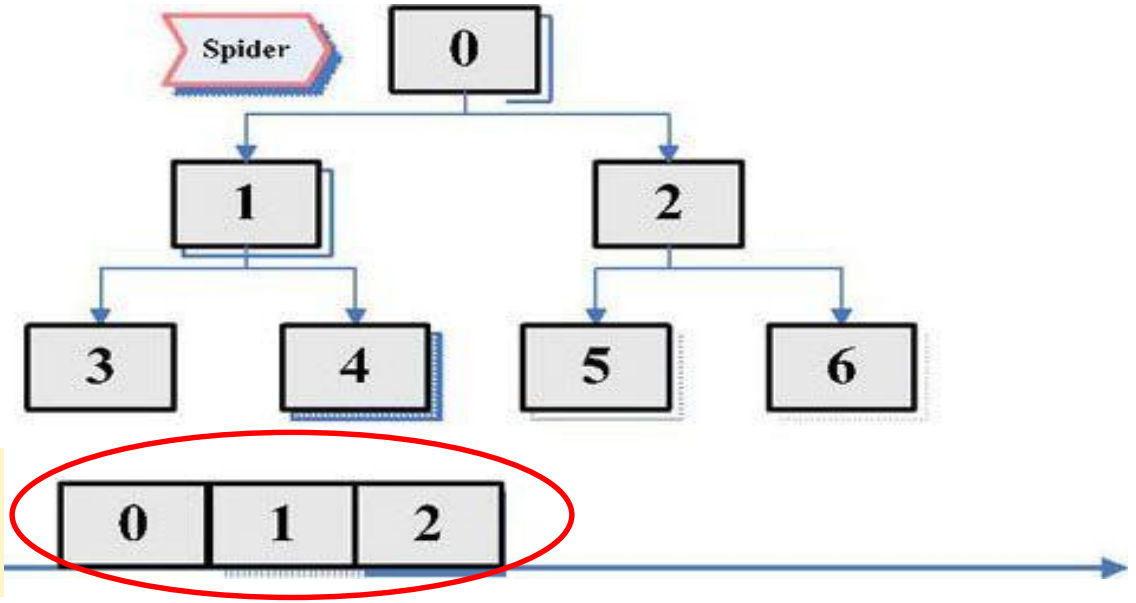


2.2.2 网页抓取原理

- 宽度优先抓取网页过程：

（1）爬虫提取负载队列中的网页0（种子站点），抓取后首先对网页0进行链接分析

将提取的网页1和网页2的链接放到待抓取工作队列中。



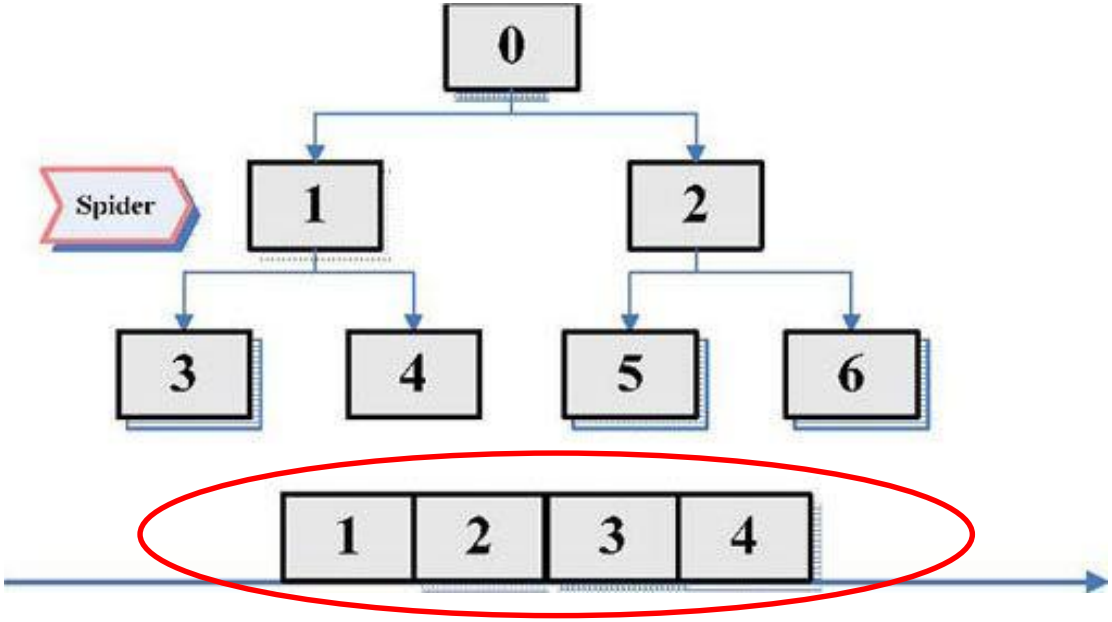
此时当前工作负载队列中增加了网页1和网页2

2.2.2 网页抓取原理

- 宽度优先抓取过程：

（2）负载队列中的网页0抓取完毕，继续抓取工作队列中第1个任务，即网页1

并提取指向网页3和网页4的链接放到工作队列中

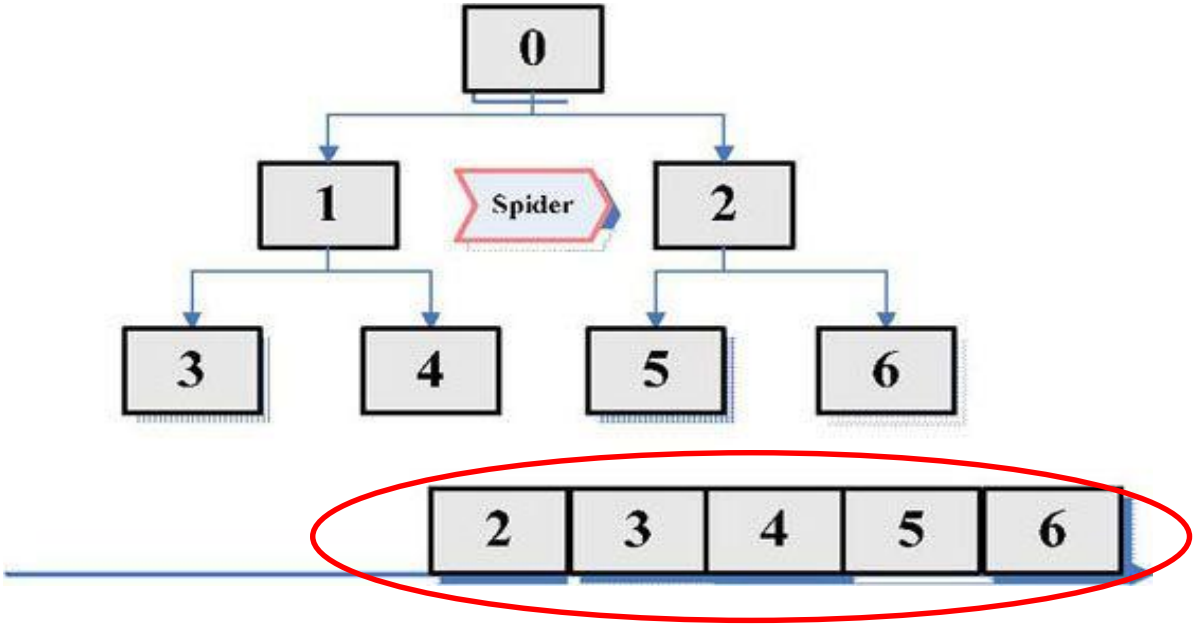


2.2.2 网页抓取原理

- 宽度优先抓取过程：

（3）根据宽度优先的规则，抓取工作队列中的网页2，并且分析出

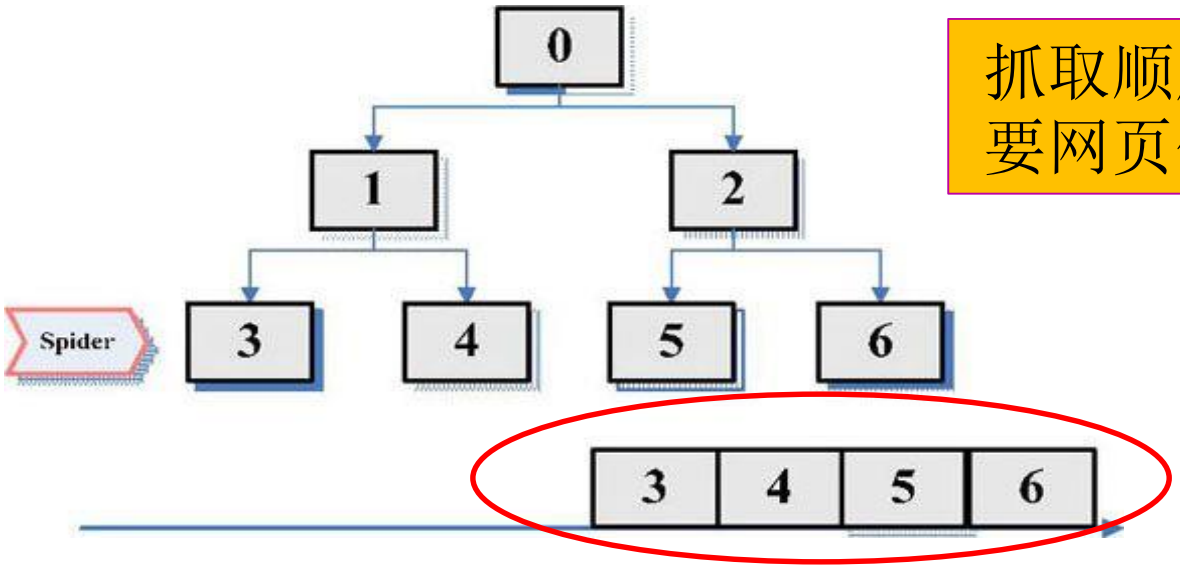
指向网页5和网页6的链接放入工作队列中



2.2.2 网页抓取原理

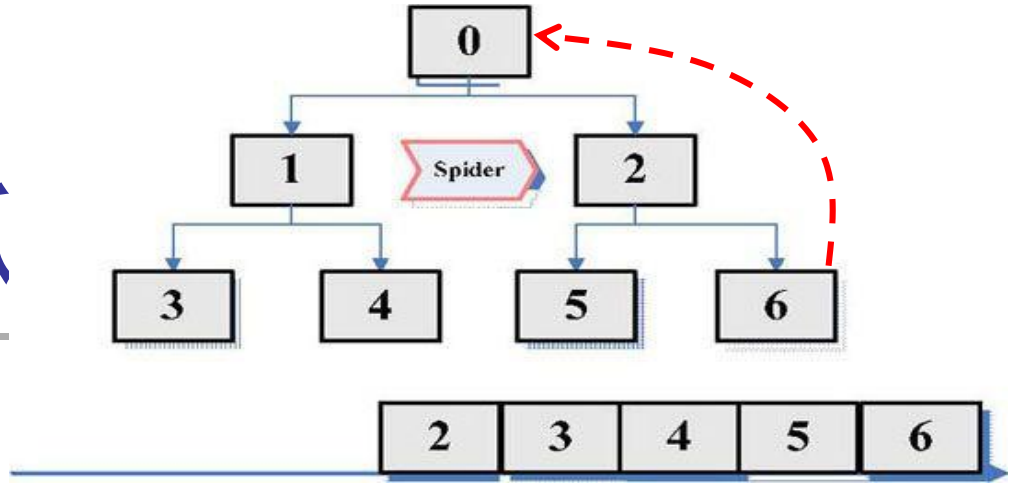
- 宽度优先抓取过程：

（4）继续抓取工作队列中的网页3，直到结束



抓取顺序恰好符合了重要网页优先抓取的要求

2.2.2 网页抓



- 宽度优先抓取问题
  - 万维网中存在大量环路，实际上网页6可能存在指向网页0的链接。
  - 如果没有任何判断，则爬虫永远抓取不完，因为存在了0、2和6这样一个死循环。
  - 死循环会带来下面两个不利后果

    （1）不该抓的反复抓。

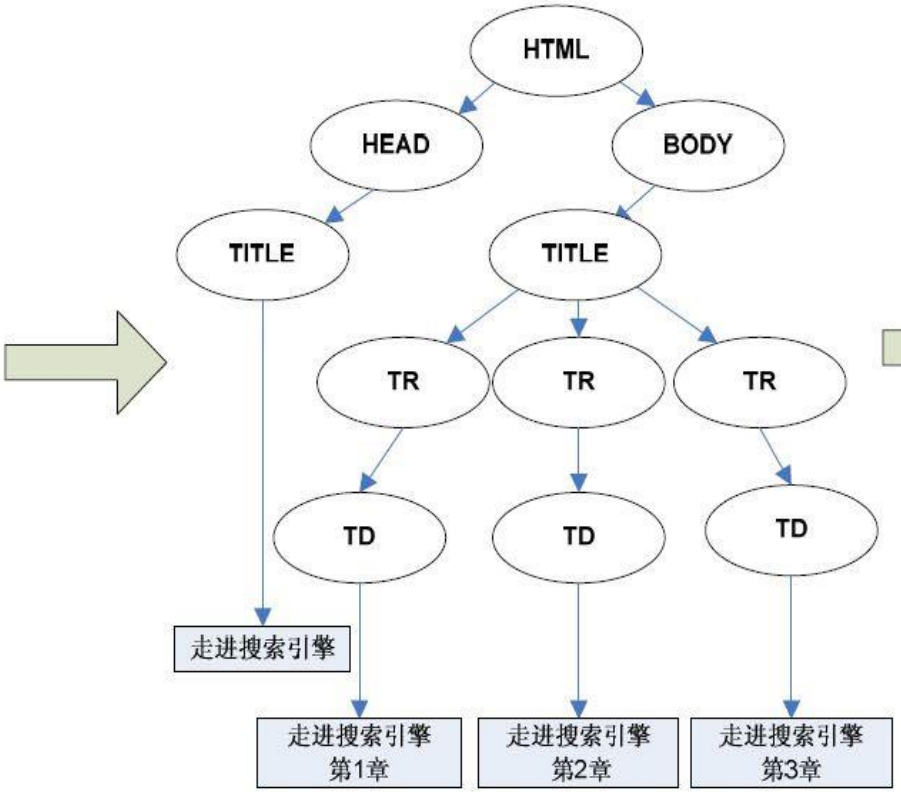
    （2）该抓的没有机会抓。

不重复抓取策略

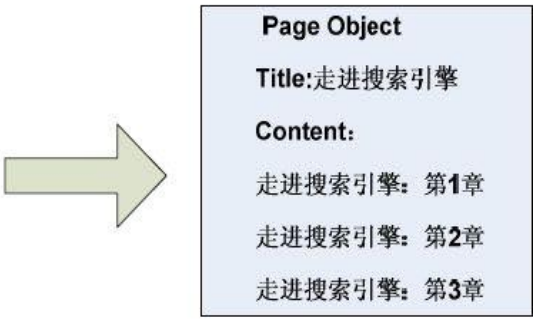
- 网页结构化完整过程（2021.3.18回顾）

```
<HTML>
<HEAD>
<TITLE>走进搜索引擎</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD>走进搜索引擎：第1章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第2章</TD>
</TR>
<TR>
<TD>走进搜索引擎：第3章</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

1）建立标签树



2）投票获得正文



- 网页结构化意义：
  - 节约大量的存储
  - 保留网页有价值的信息，例如标题和正文；

2021.3.18回顾2.2

2.2 网页信息抽取（2章）

2.2.1 万维网结构特点2.2

2.2.2 网页抓取原理2.4

1.网页抓取几个概念

2.网页获取关键问题（遍历策略、抓取顺序）

3.深度优先

4.宽度优先

5.宽度优先抓取网页

6.不重复抓取

2.2.3 网页查重3.3

2.2.4 内容扩展：Jaccard系数

2.2.2 网页抓取原理

6.不重复抓取策略

- 不重复的关键在于记住历史
- 爬虫记录历史的方式是哈希表(也称为“杂凑表”)

- **概念：**
  - 哈希表及散列函数
  - MD5签名函数
  - 记录网页访问历史

2.2.2 网页抓取原理

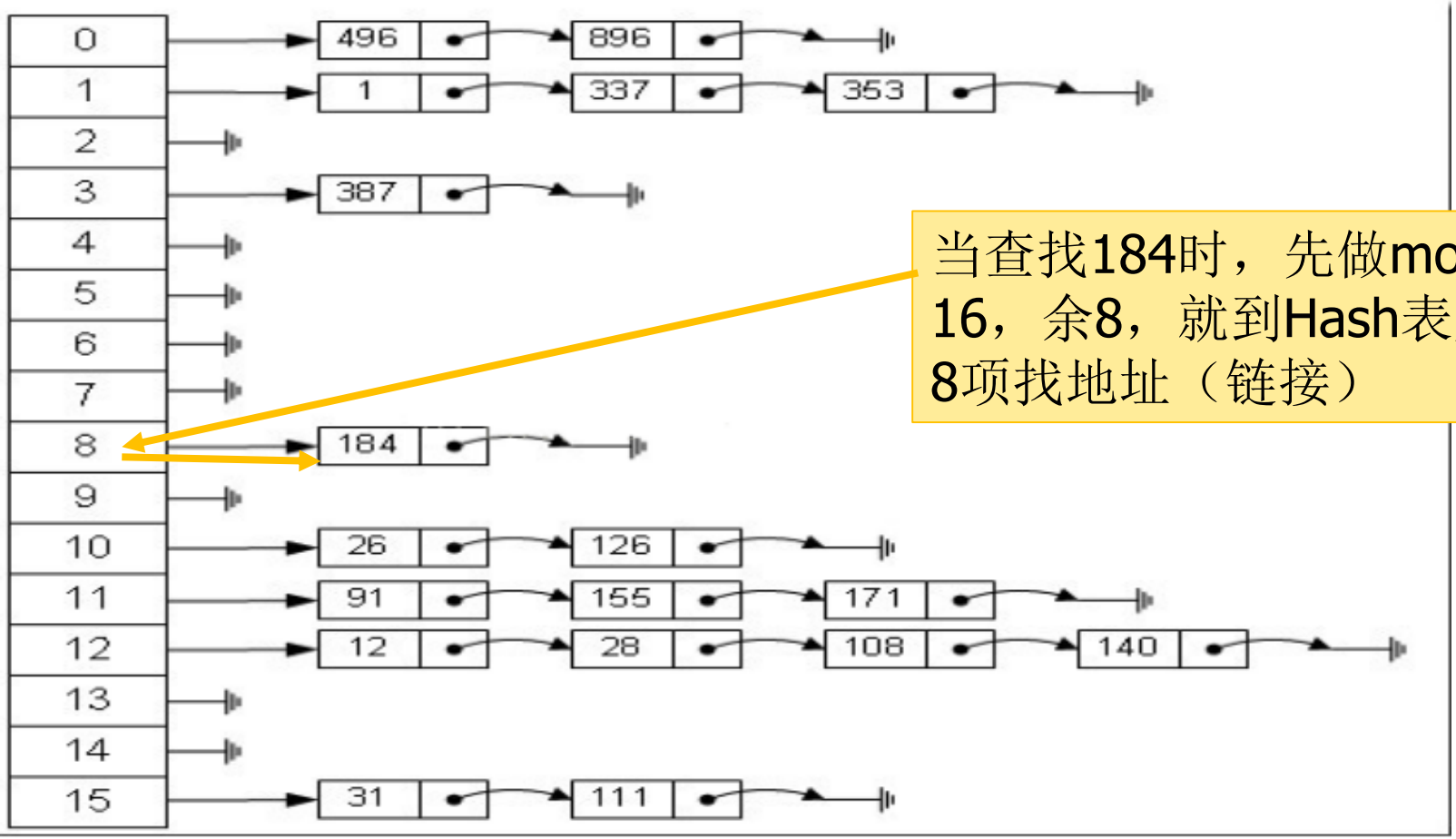
- 散列思想
  - 可以根据关键码值(Key value)直接访问的数据结构
  - 通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。
- 散列函数（哈希函数）
  - 把关键码值映射到表中一个位置的映射函数
- 散列表（哈希表）
  - 理想散列表（哈希表）是一个包含关键字的具有固定大小的数组

100个同学，学号是由院系-年级-班级和编号组成，成绩建表（例01100168表示是1系，10级1班的68号）
为了快速查找到68号的成绩信息，用学号作为下标数值实在太大。
方法：将学号除以1100100取余，即得到编号作为该表的下标，直接访问表下标为68的数据即可。
这就能够在O（1）时间复杂度内完成成绩查找。

2.2.2 网页抓取原理

怎样使不同的值映射为唯一地址？

- 例.我们定义哈希函数H(x) = x mod 16



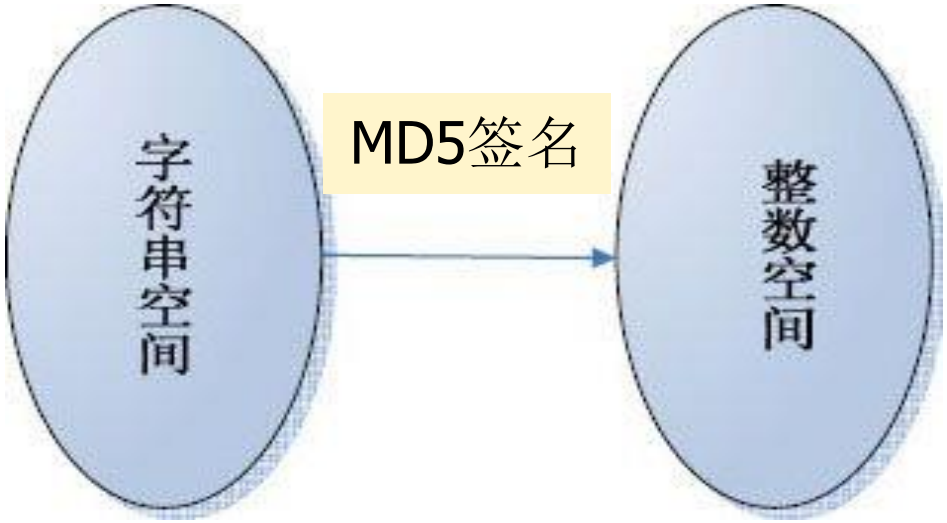
当查找184时，先做mod 16，余8，就到Hash表第8项找地址（链接）

2.2.2 网页抓取原理

- MD5签名函数
  - 散列函数，可以将任意长度的数据流转换为一个固定长度的数字(通常为4个整型数，即128位)
  - MD5签名后的数字称为“数据流的签名”或者“指纹”
  - 数据流中的任意一个微小的变化都会导致签名值发生变化。
- 记录网页访问历史
  - 散列表的值记录网页抓取情况：如果某网页在过去的某个时刻已经被抓取，置1，反之置0
  - 具体映射到哪一个槽位，则由哈希函数决定。

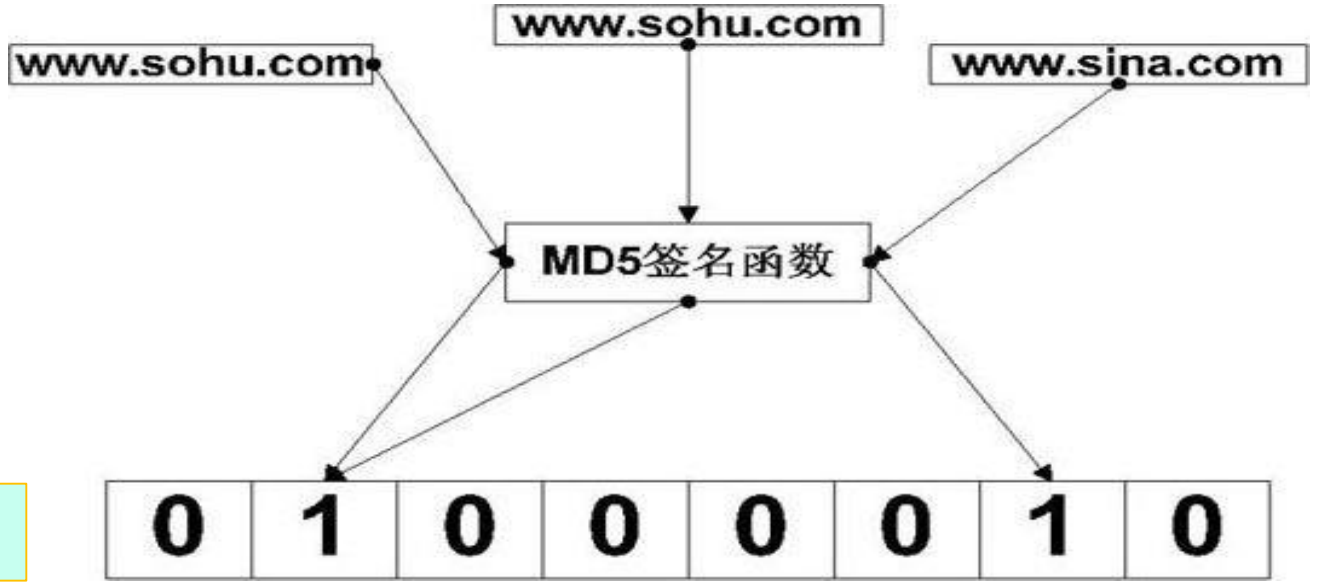
2.2.2 网页抓取原理

- 记录网页访问历史过程：
  - 将网页标识（URL字符串）数字化：唯一地计算成一个整数。
  - 在上面整数映射为唯一一个整数（散列地址）
  - 在散列地址对应的散列表的槽位记录网页访问状态



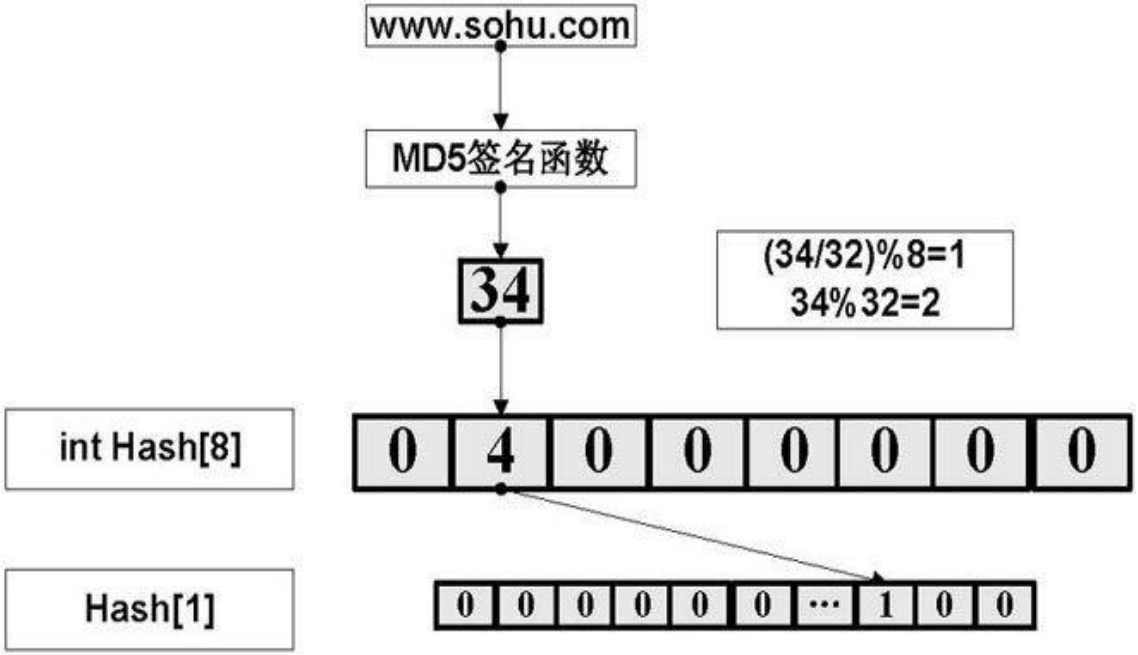
2.2.2 网页抓取原理

- 网址唯一签名例



2.2.2 网页抓取原理

- 哈希表实际应用须解决的其他问题：
  - 哈希数组尽可能地全部放在内存中，以保证每一个网页的签名都可以通过查找哈希表来确定是否曾经抓取过
  - 采用Bitmap的数据结构压缩内存用量
  - Bitmap需要借助按位与（&）和按位或（|），以及左移（<<），右移（>>）这4种基本位运算



第2章 万维网网页信息的表达及解析

2.2 网页信息抽取（2章）

2.2.1 万维网结构特点2.2

2.2.2 网页抓取原理2.4

2.2.3 网页查重3.3

2.2.4 内容扩展：Jaccard系数

2.2.3 网页查重

- 对搜索引擎来说，重复网页无论从系统效率，还是从检索质量来说，都是有害的问题，具体：
  - 这些网页至少被多处理一次
  - 在索引制作中可能会在索引库中索引两份相同的网页
  - 当有用户查询时，在有限的查询结果页中（一个查询结果页显示20条网页链接）就会出现重复的网页链接

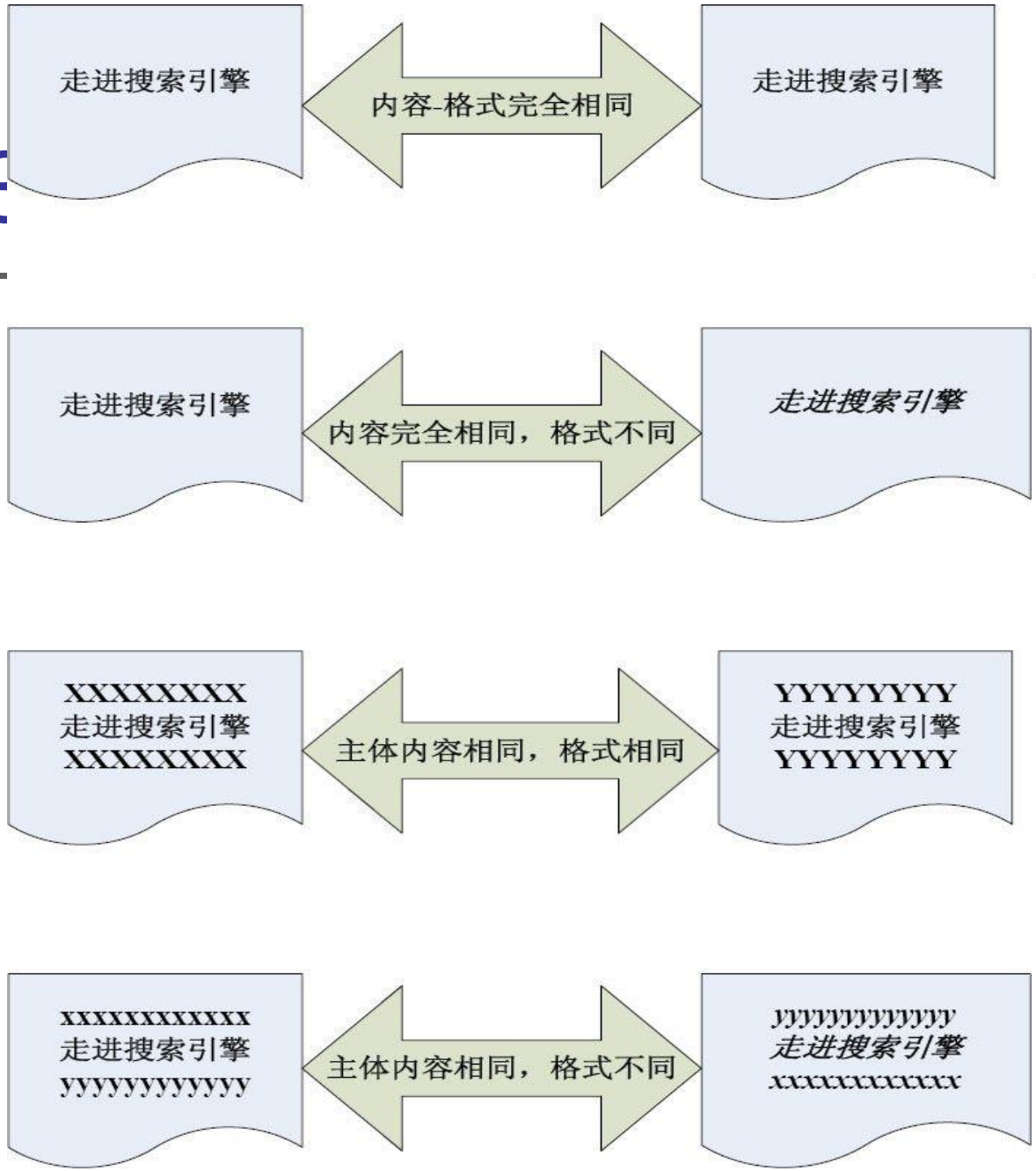
搜索引擎中需要网页查重

2.2.3 网页查重

网页查重技术发展历史

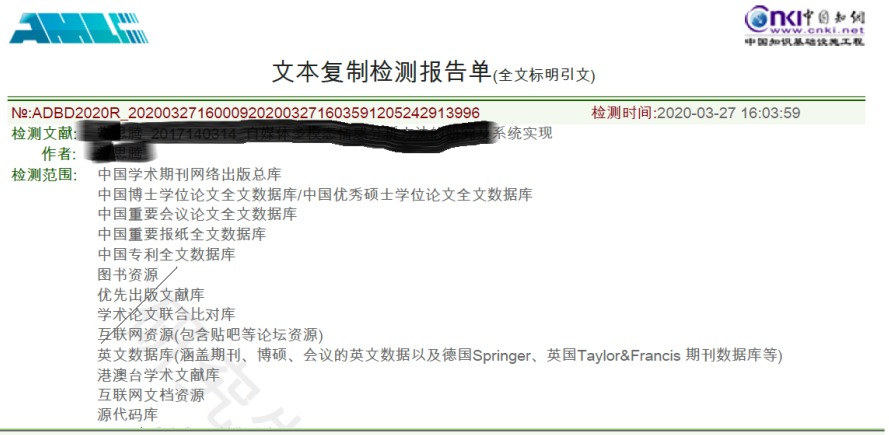
- 起源于复制检测技术，即判断一个文件的内容是否抄袭、剽窃或者复制于另外一个或者多个文件。
- 1993年，Arizona大学的Manber推出一个sif工具，用于在大规模文件系统中寻找内容相似的文件。
- 1995年，Stanford大学的Brin和Garcia-Molina等人在“数字图书馆”工程中首次提出了文本复制检测机制COPS（Copy Protection System）系统与相应算法
- 之后这种检测重复技术被应用到搜索引擎中，基本的核心技术即比较相似。

网页由内容和格式决定，4种网页相似的类型：



2.2.3 网页查重

- 网页查重方法
  - 检测网页完全一致



文本复制检测报告单(全文标明引文)

№:ADBD2020R_20200327160009202003271603591205242913996　　检测时间:2020-03-27 16:03:59

检测文献: [redacted]系统实现

作者: [redacted]

检测范围:
中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
学术论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
源代码库

相似文献列表

去除本人已发表文献复制比：9.2%(530)　　文字复制比：9.2%(530)　　疑似剽窃观点：(0)

1	基于微博话题的情感分析模型的研究 何志强(导师：李文敏) - 《北京邮电大学硕士论文》- 2019-02-20	6.7%（388） 是否引证：是
2	基于层次化双向LSTM的评论方面级别情感分析研究 孟霞(导师：左万利) - 《吉林大学硕士论文》- 2019-05-01	1.4%（79） 是否引证：否
3	船舶监控系统中无线传感器网络数据融合关键技术研究 张峰(导师：郑洪源) - 《南京航空航天大学硕士论文》- 2017-06-01	1.2%（70） 是否引证：否
4	基于改进图割算法的脑部MRI分割技术研究 田换(导师：元昌安) - 《广西师范学院硕士论文》- 2016-06-01	1.1%（62） 是否引证：否
5	红外运动小目标检测方法综述 田鹏辉;隋立春;燕莎; - 《探测与控制学报》- 2013-04-26	0.9%（50） 是否引证：否
6	冰球教练辅助系统的研究与实现 肖雪(导师：薛善良) - 《南京航空航天大学硕士论文》- 2019-03-01	0.6%（35） 是否引证：否
7	一种基于瓦记录磁盘的键值存储系统的研究与实现 叶虎(导师：谢长生;万继光) - 《华中科技大学硕士论文》- 2019-05-01	0.6%（35） 是否引证：否

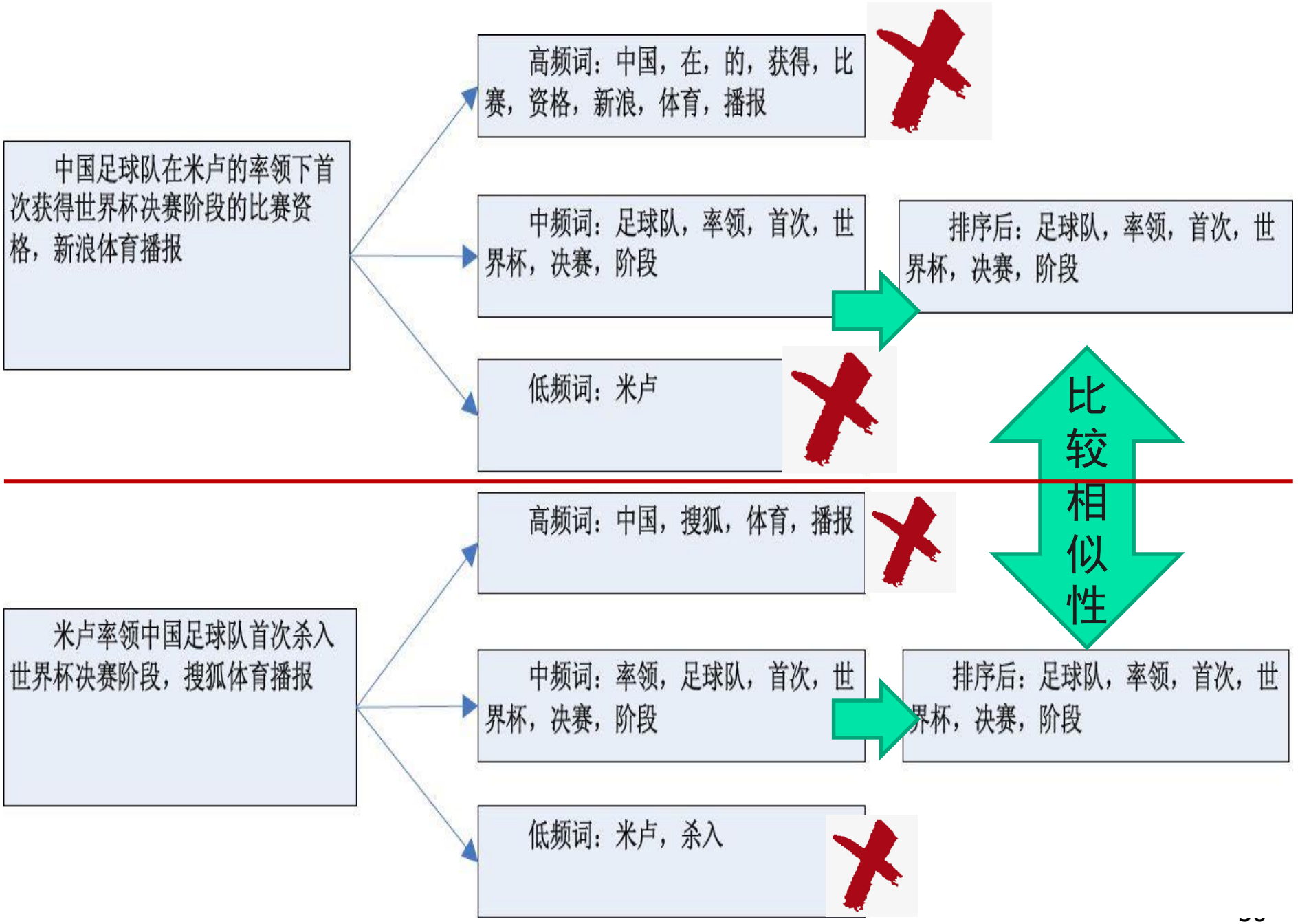
2.2.3 网页查重

- 文档查重：只考虑网页内容
  - 比较“标题+正文文档”
    - 不考虑格式的异同，首先需提取结构化网页的正文和标题，将复杂的网页转化为具有标题和正文的文档
  - 方法：比对“特征”
    - 特征是那些有代表性、不易变化、能够反应主体本质的信息。
    - 特征抽取的方法：
      - I-Match算法
      - Shingle算法

2.2.3 网页查重

- I-Match算法
  - 算法基于一个假设，即一篇文档中特别高频和特别低频的词汇往往不能反映这篇文档的本质
    - 去掉高频和低频词汇
    - 排序得到一个字符串
    - 使用签名算法获得该字符串的签名
    - 如果有其他文档和这个签名值相同，则判定为相似
  - 尽可能抽取一个特征，这样比较两个文档是否相似只要比较一次即可

不依赖于完全的信息分析，使用数据集合的统计特征来抽取文档的主要特征，将非主要特征抛弃。

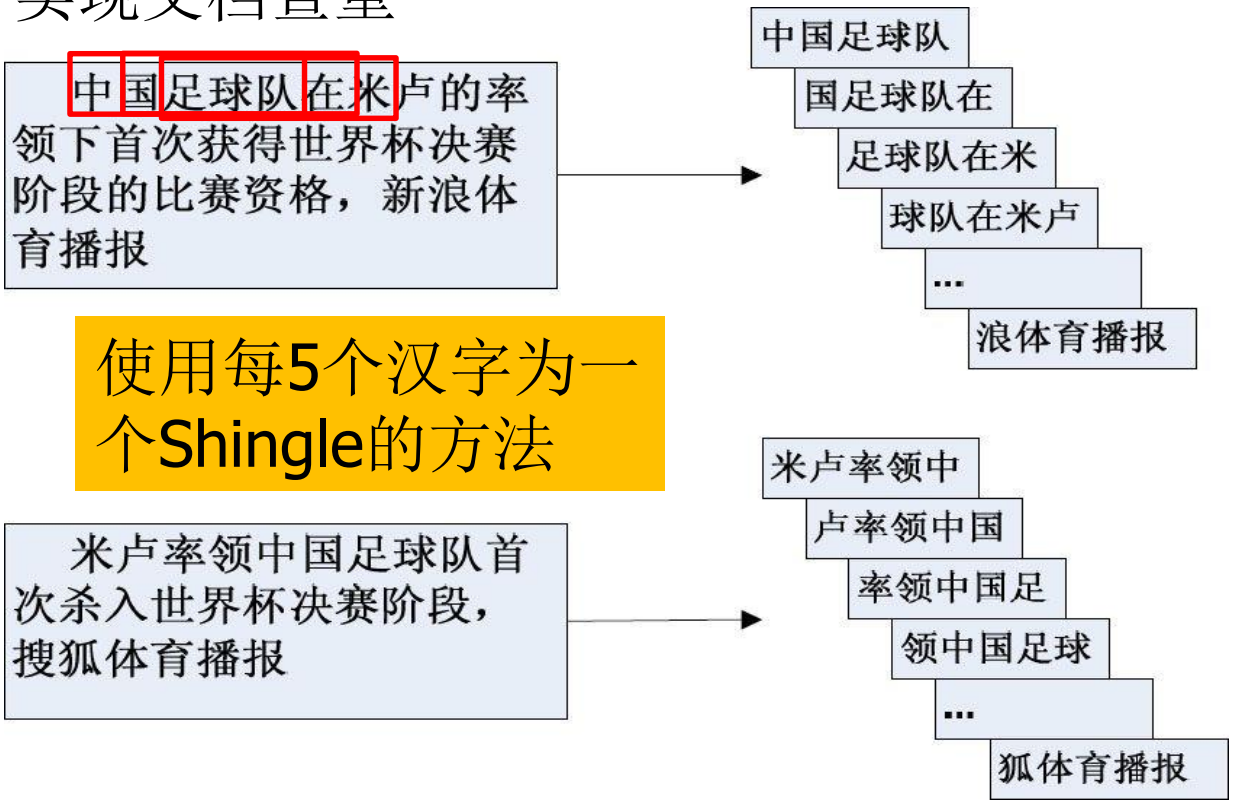

中国足球队在米卢的率领下首次获得世界杯决赛阶段的比赛资格，新浪体育播报
高频词：中国，在，的，获得，比赛，资格，新浪，体育，播报
中频词：足球队，率领，首次，世界杯，决赛，阶段
排序后：足球队，率领，首次，世界杯，决赛，阶段
低频词：米卢
比较相似性
米卢率领中国足球队首次杀入世界杯决赛阶段，搜狐体育播报
高频词：中国，搜狐，体育，播报
中频词：率领，足球队，首次，世界杯，决赛，阶段
排序后：足球队，率领，首次，世界杯，决赛，阶段
低频词：米卢，杀入

2.2.3 网页查重

- Shingle算法（相互覆盖的瓦片）
  - 原理：
    - 把一个文档转化为一组字符串集合（每个元素为一个Shingle），将判断两个文档的相似性转化为字符串集合的相似性。
  - 步骤：
    - 1.确定Shingle：对于长度L的文档，每隔N个汉字取一个Shingle，这样一共取到L－N＋1个Shingle，可见N的取值对效率和效果的影响都很大。很显然N最小取2，最大取L。
    - 2.提取特征（Shingle集合）
    - 3. 特征比对--相似度计算和评价。

2.2.3 网页查重

- Shingle算法（相互覆盖的瓦片）
  - 抽取多个特征词汇，通过比较两个特征集合的相似度实现文档查重



2.2.3 网页查重

- I-Match算法与Shingle算法比较：
  - I-Match算法提取特征时需要文本分词和词频比较的代价，因此提取特征比较复杂，但文档是否相似的计算简单；
  - Shingle算法提取特征简单，但文档是否相似的计算复杂，因此各有利弊。由于Shingle算法在性能上的优异表现，因而被广泛采用。
- 这里介绍的是最经典的两种方法，其他网页查重的方法还有很多。

2.2.3 网页查重

- 网页消重
  - 网页查重的第3步就是网页消重。
  - 消重，几个角度：
    - 从版权的角度考虑，应该尊重原创，过滤转载或者复制的网页；
    - 从网页寿命的角度考虑，过滤掉那些网站质量不高的网页，保留大型网站的网页；
    - 从容易实现的角度考虑，首先保留被爬虫抓取的网页，然后丢弃被抓取的相同或相似网页——最为简单实用，由于保留先被爬虫抓取的网页同时很大程度上也保证了优先保留原创网页的原则，因此被广泛采用。

2.2.3 网页查重

- 网页消重3个主要步骤：
  - （1）特征抽取。
  - （2）相似度计算、评价是否相似。
  - （3）消重。
- 网页查重工作是分析系统基础工作，为接下来的网页评价，节约PageRank的计算代价、节省了索引存储空间、减少了查询成本、提高了查询结果的多样化效果。

第2章 万维网网页信息的表达及解析

2.2 网页信息抽取（2章）

2.2.1 万维网结构特点2.2

2.2.2 网页抓取原理2.4

2.2.3 网页查重3.3

2.2.4 内容扩展：Jaccard系数

2.2.4 内容扩展：Jaccard系数

例：判断以下两个网页是否相似

采用Shingle法提取网页特征

明起电话订火车票可全国通取取票时间延12小时

明起 起电 电话 话订 ..... 小时

明起、起电、电话、话订、订火、火车、车票、票可、可全、全国、国通、通取、取取、取票、票时、时间、间延、延12、12小、小时

（L-N+1=21-2+1=20个）

共同的Shingle7个

火车票电话订票实现全国通取网上预售期延长

火车 车票 票电 电话 ..... 延长

火车、车票、票电、电话、话订、订票、票实、实现、现全、全国、国通、通取、取网、网上、上预、预售、售期、期延、延长

（L-N+1=20-2+1=19个）

文档标题的相似度7/(20+19-7)=0.21875——Jaccard系数

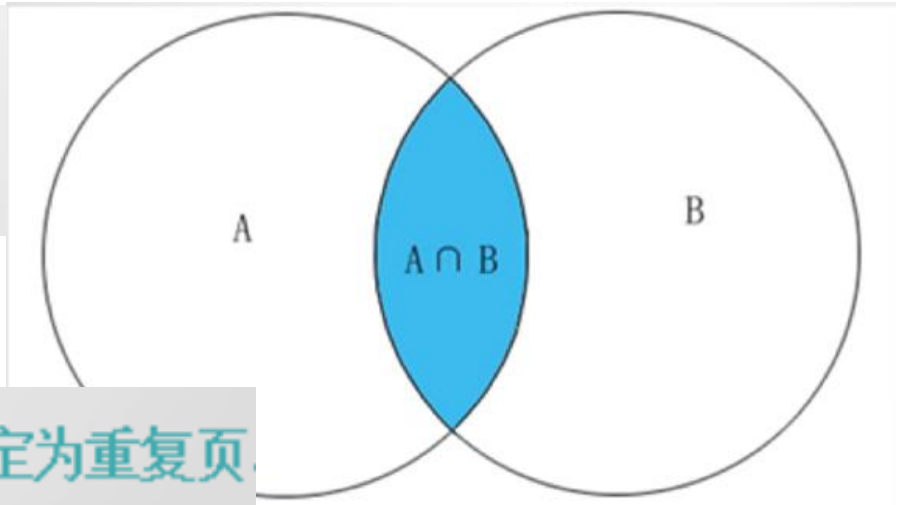
扩展——Jaccard系数

- Jaccard系数
  - 给定两个集合A和B，A和B的

Jaccard相似度 =

|A与B的交集元素个数| / |A与B的并集元素个数|

$$Jaccard系数=\frac{A \cap B}{A \cup B}$$



计算jaccard系数大于某个数，就判定为重复页

扩展——Jaccard系数、 Jaccard距离

- Jaccard 系数(Jaccard index，又称Jaccard相似系数）
  - 用于比较有限样本集之间的相似性与差异性。
  - Jaccard系数值越大，样本相似度越高。

主要应用场景:比较文本相似度，用于文本查重与去重

- Jaccard 距离
  - 与Jaccard 系数相关的指标
  - 用于描述集合之间的不相似度。
  - Jaccard 距离越大，样本相似度越低。

主要应用场景:计算对象间距离，用于数据聚类等

扩展——Jaccard系数应用示例

Jaccard相似度在竞品分析中的应用

https://www.cnblogs.com/charlotte77/p/7504251.html

- 项目场景：竞品分析——**分析与博客园竞争的同类技术网站相似度——可以同类化于相似度的计算问题**
- 相似度计算方法很多，常见的欧几里得距离，余弦计算，皮尔逊距离，Jaccard相似度计算
- 人类经验总结：
  - 喜欢博客园的用户也喜欢浏览知乎、CSDN、Github等
  - 喜欢知乎的用户也喜欢浏览Github、InfoQ、V2EX、SegmentDefault、博客园等
- 解题：我们根据两类用户浏览相近网站次数来进行排序，得到两个集合，比较它们的相似度

博客园=[知乎、CSDN、Github]

知乎=[Github、InfoQ、V2EX、SegmentDefault、博客园]

Jaccard相似度为= 1 / 7=0.14
——不太合理

扩展——Jaccard系数应用示例

Jaccard相似度在竞品分析中的应用
https://www.cnblogs.com/charlotte77/p/7504251.html

- 调整竞品集合：
  - 将需要计算的竞品本身也加入集合

博客园=[博客园、知乎、CSDN、Github]

知乎=[知乎、Github、InfoQ、V2EX、SegmentDefault、博客园]

博客园与知乎的Jaccard相似度 = 3 / 7 = 0.42

- 为什么我们要将竞品本身考虑进去呢？
  - 我们的目的是找到博客园的竞品，分析出经常浏览博客园的用户还会经常浏览哪些同类技术网站，那么博客园的用户肯定是经常浏览博客园的，这点显而易见，一个物品本身也是自身的竞品
  - 将要分析的竞品本身加入集合后就可避免我们第一次计算时出现的不符合常识的结果。

- 思考一个问题：
  博客园对知乎的Jaccard相似度与知乎对博客园的Jaccard相似度应该是一样的吗？
  - 按照前两次计算，我们认为是一样的，因为只是考虑到交集的个数，并没有考虑集合中元素所处的位置因素。
  - 然而实际上，集合中的元素位置其实是有先后之分的，按降序排列，即竞品相关度是越来越低的。
  - 举个例子：一个经常看博客园的用户，也会经常看知乎，那么一个经常看知乎的用户是否也代表也会经常看博客园呢？
    - 这个结论与我们给出的条件是相悖的：一个经常看知乎的用户，相比于博客园，更偏好于Github。
    - 所以我们得到结论：两个竞品A和B，A对B的重要性不一定等于B对A的重要性。

博客园=[博客园、知乎、CSDN、Github]
[1.0,0.6,0.3,0.1]

知乎=[知乎、Github、InfoQ、V2EX、SegmentDefault、博客园]
[1.0,0.55,0.15,0.14,0.11,0.05]

给竞品排位置、加权重

- 博客园对知乎的Jaccard相似度 =

（ 两者交集的权重得分和/ 两者权重总和 ）* 知乎在博客园集合中所占的权重 =（ 1+0.6+0.1+1+0.55+0.05 / (2+2) ）* 0.6 = （ 3.3 /4 ）* 0.6 = **0.495**

博客园与知乎的竞品相似度是不相同的-符合常理

- 知乎对博客园的Jaccard相似度 =

（ 两者交集的权重得分和/ 两者权重总和 ）* 博客园在知乎集合中所占的权重 =（ 1+0.6+0.1+1+0.55+0.05 / (2+2) ）* 0.05 = （ 3.3 /4 ）*0.05 = **0.04**

- 针对每一次计算的结果，结合常识，再来进行一步步改进，最后取得了不错的效果。其实最后的方案还可以做一些改进
- 如：如何设定权重，如何设定计算公式、是否可以用线性模型拟合、以及最后乘以的权重如果影响太大，是否可以改成根据位置进行指数衰减等等
- 都可以去尝试，有兴趣的也可以去试一试。