# 题型

简答：3×5′

仅供参考↓

综合题：4x20′+1x15′

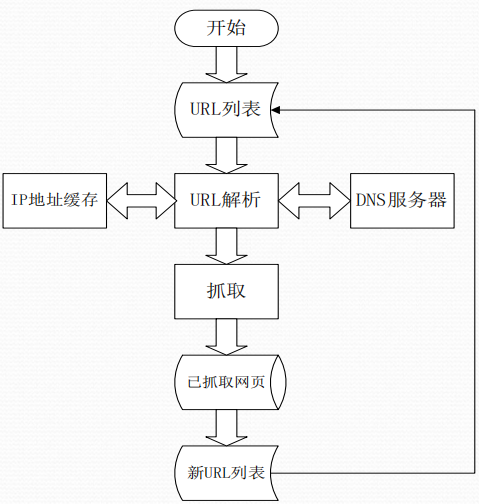
# Chapt3 网页抓取技术

## 爬虫概念

网络爬虫（Crawler），也称为蜘蛛程序（Spider），或网络机器人（Robots）。网络爬虫是一个自动提取网页的程序，尽可能多和快的给索引部分输送网页。通过网页的链接地址来寻找网页，从网站某一个页面开始，读取网页的内容，找到在网页中的其它链接地址，然后通过这些链接地址寻找下一个网页，一直循环下去，直到把这个网站所有的网页都抓取完为止。

## 爬虫工作原理/基本流程

图例（助记）



原理/流程

在爬虫开始的时候，需要给爬虫输送一个URL列表，列表中的URL地址就是爬虫的起始位置。爬虫从这些URL出发，开始爬行，一直不断地发现新的URL，然后再根据策略爬行这些新发现的URL，如此永远反复下去。一般的爬虫都自己建立DNS缓冲，建立DNS缓冲的目的是加快URL解析成IP地址的速度。

## 广/深度优先策略

深度优先策略是尽量往最远的地方走，直到不能再走为止；使用该策略容易爬行很多重复的结点，需要能控制爬行路径、避免路径重复的算法。

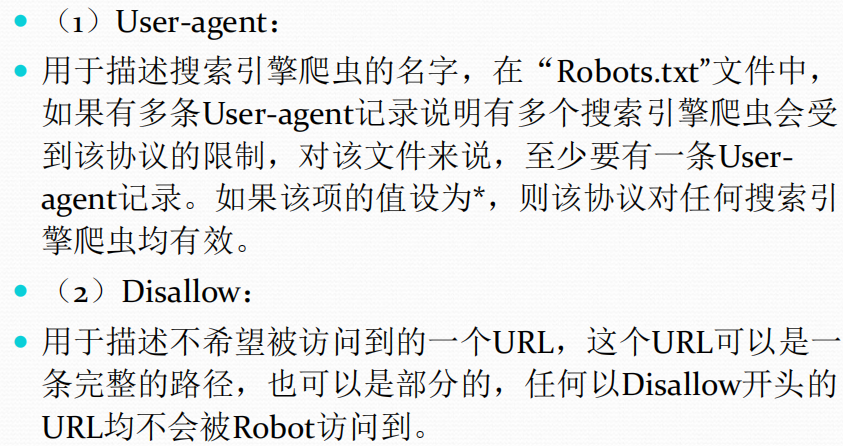
广度优先策略是一种层次型距离不断增大的遍历方法。广度优先更适合爬虫的分布式处理，启动多个爬虫，每个爬虫负责一层。此外，重要的网页往往离种子站点距离较近，随着不断的浏览，所看到的网页的重要性越来越低。

## Robots协议是什么

### 协议内容

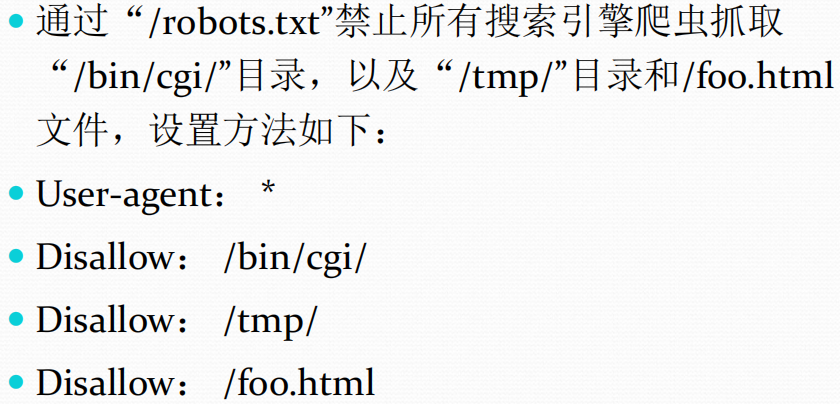
Robots协议是Web站点和搜索引擎爬虫交互的一种方式，Robots.txt文本文件规定爬虫只抓取指定内容，或禁止爬虫抓取网站的某些内容。当一个爬虫访问一个站点时，它会首先检查该站点根目录下是否存在robots.txt，如果存在，爬虫就会按照该文件中的内容来确定访问的范围；不存在，爬虫就沿着链接抓取。

### 使用格式

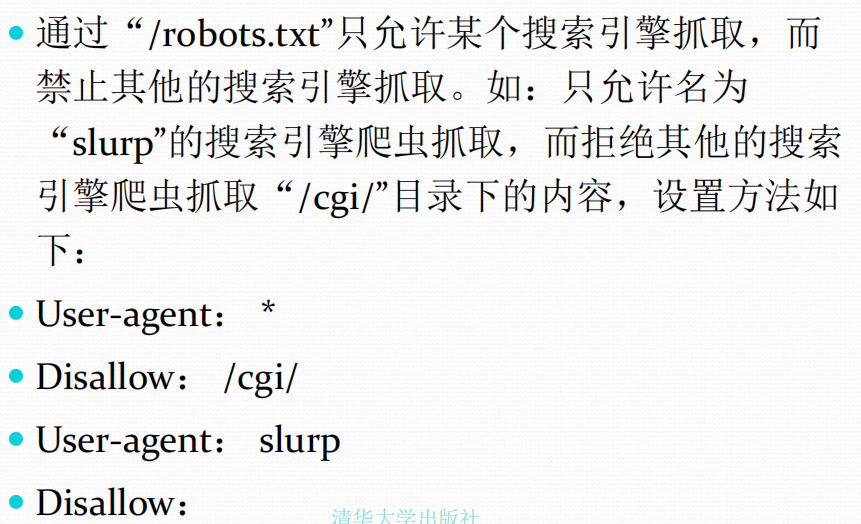


### 例题

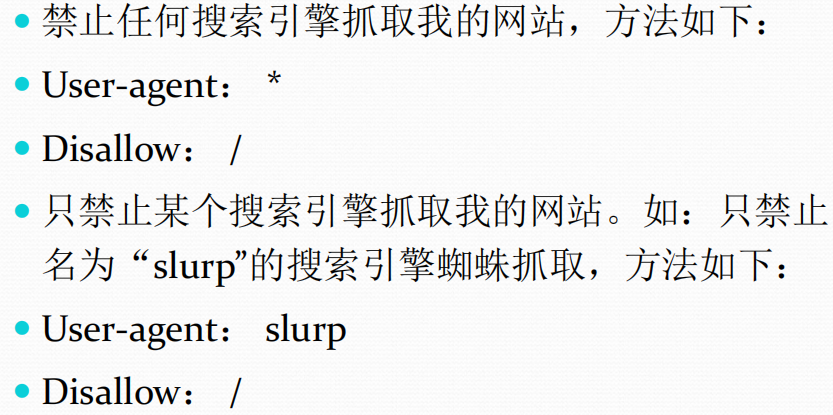
例题1



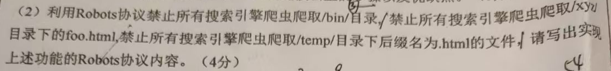
例题2

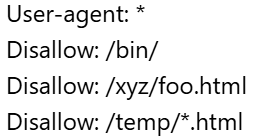


例题3



例题4





# Chapt4 网页爬虫开发技术

## 正则表达式

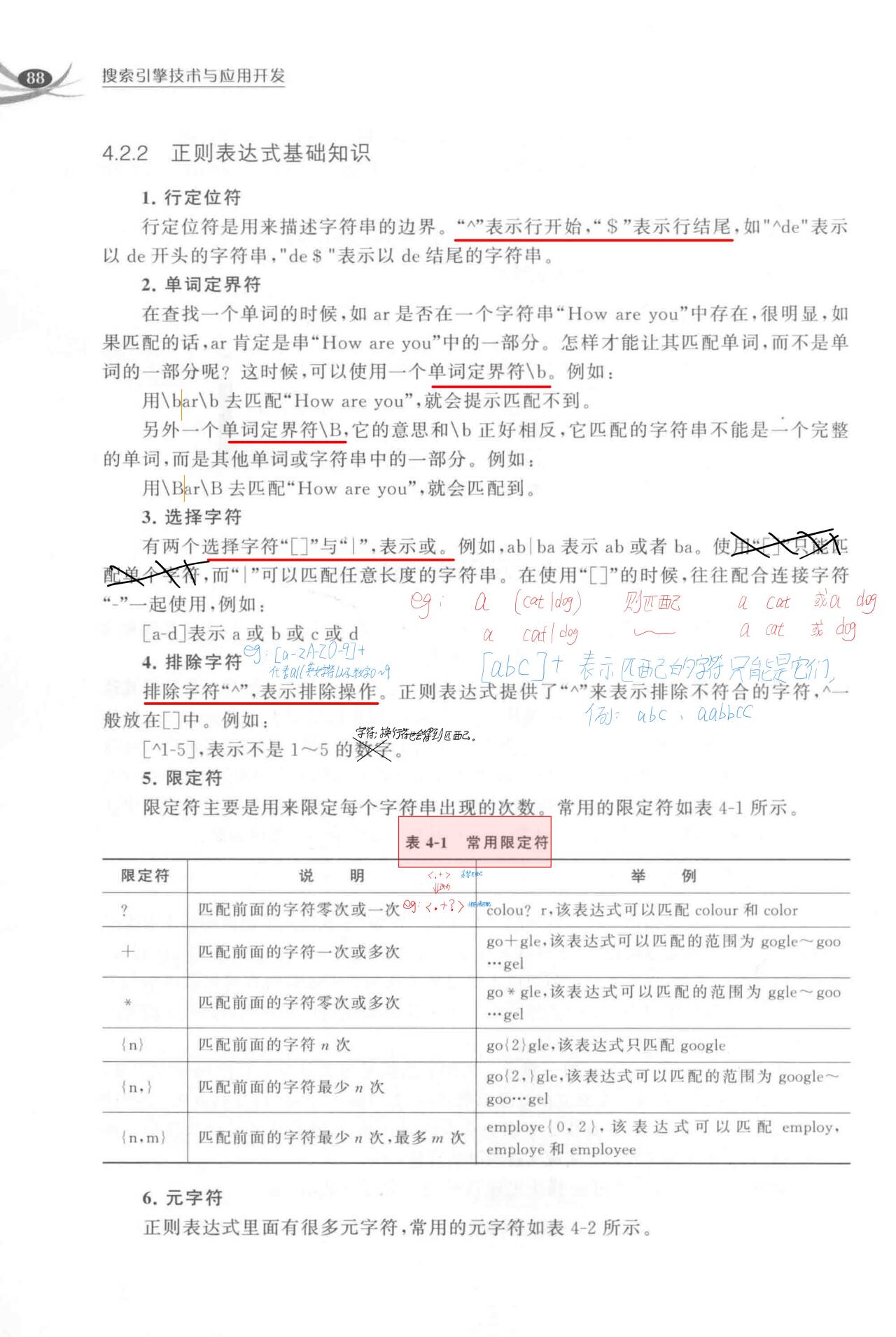
### 概念

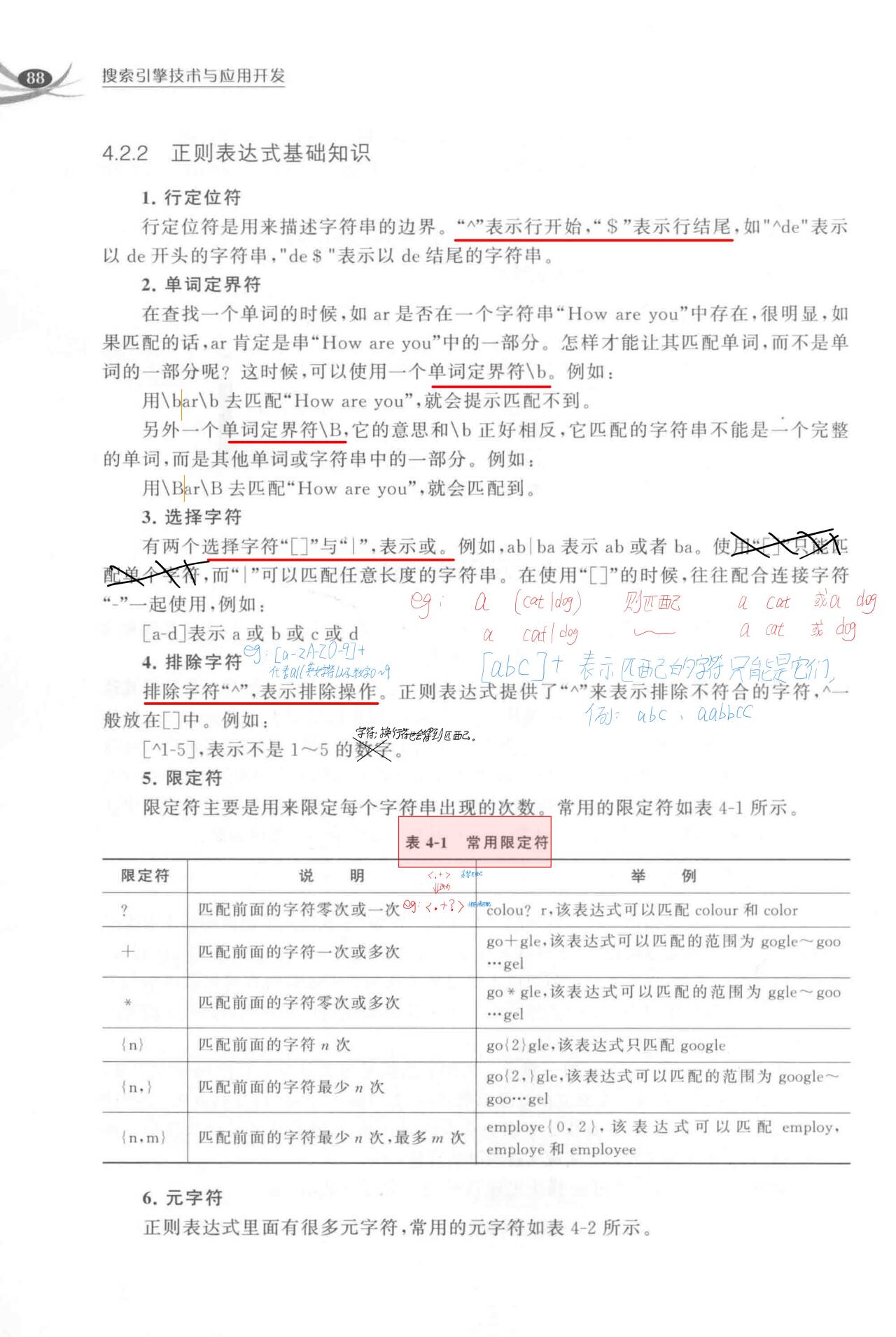
正则表达式是一个特殊的字符序列，用于检查一个字符串是否与某种模式匹配。

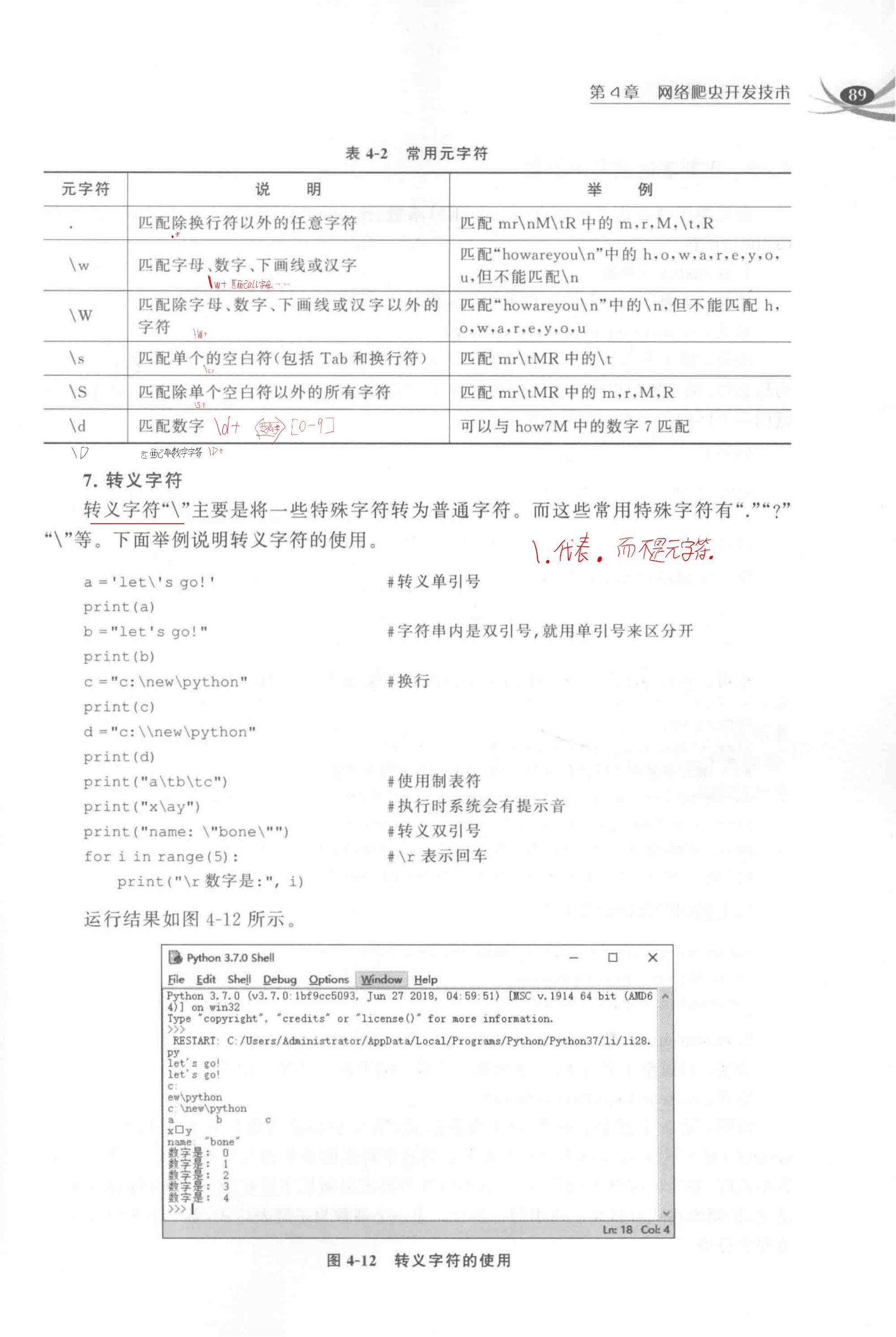
是描述字符串排列的一套规则。

是用于处理字符串的工具，拥有独特的语法以及一个独立的处理引擎，功能强大。

### 语法







### 四个模块函数

### 例题

## 爬虫框架（了解）

|  |  |
| --- | --- |
| Scrapy | 爬虫框架 |
| Crawley | 爬虫框架 |
| PySpider | 爬虫框架 |
| Portia | 爬虫框架 |

# Chapt5 网页信息预处理

## DOM树（了解）

### 概念

DOM即文档对象模型，是一种将文档（如HTML、XML等）表示为树形结构的对象模型，提供了一种结构化的方式来访问和操作文档中的内容和元素。

### DOM树建立过程

①建立标签分析栈。

②顺序读取网页标签并依次入栈。

③文本结点不入栈。

④成对标签同时退栈。

DOM树建立以后，遍历树中的每个结点，将其中的文本送到分词模块进行处理。

## 文本处理

### 列举三种中文分词方法

①基于词典的分词方法（或“基于字符串匹配的分词方法”），也叫机械分词方法。

②基于统计的分词方法。

③基于理解的分词方法。

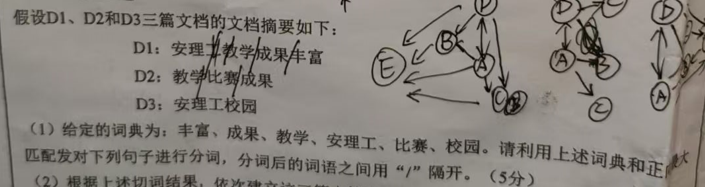
### 详述基于词典的分词方法的原理

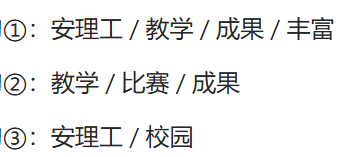
①正向最大匹配法(FMM)

算法思想

匹配方向是从左向右。选取包含6~8个汉字的符号串作为最大符号串，把最大符号串与词典中的单词条目相匹配，如果不能匹配，就减掉一个汉字继续匹配，直到在词典中找到相应的单词为止。

例题



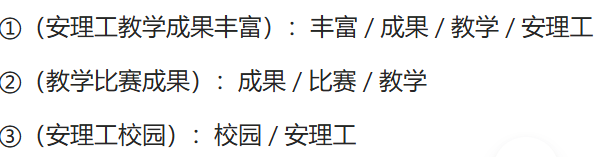


②逆向最大匹配法（BMM)

算法思想

匹配方向是从左到右。与正向最大匹配法类似。

例题



# Chapt6 信息索引技术

## 顺排检索

### 规则

最终需要画的表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址 | 检索词 | 条件满足指向 | 条件不满足指向 | 级位 |
| 1 | A | … | … | … |
| 2 | B | … | … | … |
| … | … | 命中 | 落选 | … |

填写依据

（1）填写

地址：从A到字母Xn依次为1，2，…，n。

检索词：按表达式从左到右的字母，依次填入。

级位：X、（X）的级位为0；（X 遇左括号的级位加1；X） 遇右括号的级位减1；括号内的其他情况，级位不变。

条件满足指向（AFD）：X\*，针对\*运算符，填入指向后一词的地址。

条件不满足指向（NFD）：X+，针对+运算符，填入指向后一词的地址。

Ps：最后一个检索词的AFD、NFD必为命中、落选。

（2）补全

针对AFD和NFD内容，从下往上补全，设地址为ai。

AFD

若ai的级位**＞**a(i+1)的级位，则将a(i+1)的AFD内容复制到ai的AFD内容；

否则比较ai与a(i+2)…不断向下比较，直至找到可复制的内容为止。“命中、落选”也是可复制的内容。

NFD

若ai的级位**≥**a(i+1)的级位，则将a(i+1)的AFD内容复制到ai的AFD内容；

否则比较ai与a(i+2)…不断向下比较，直至找到可复制的内容为止。“命中、落选”也是可复制的内容。

（3）观察有无负号

若有，则在-X的所在地址行，对其“条件满足指向”和“条件不满足指向”的内容添加一个“相反”的符号即可。即：

|  |  |
| --- | --- |
| … | … |

### 例题

例题1



例题2

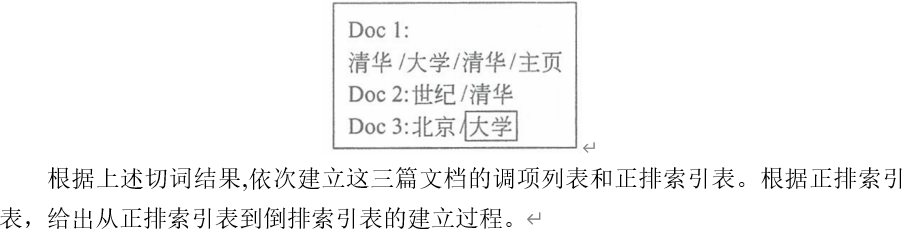


例题3

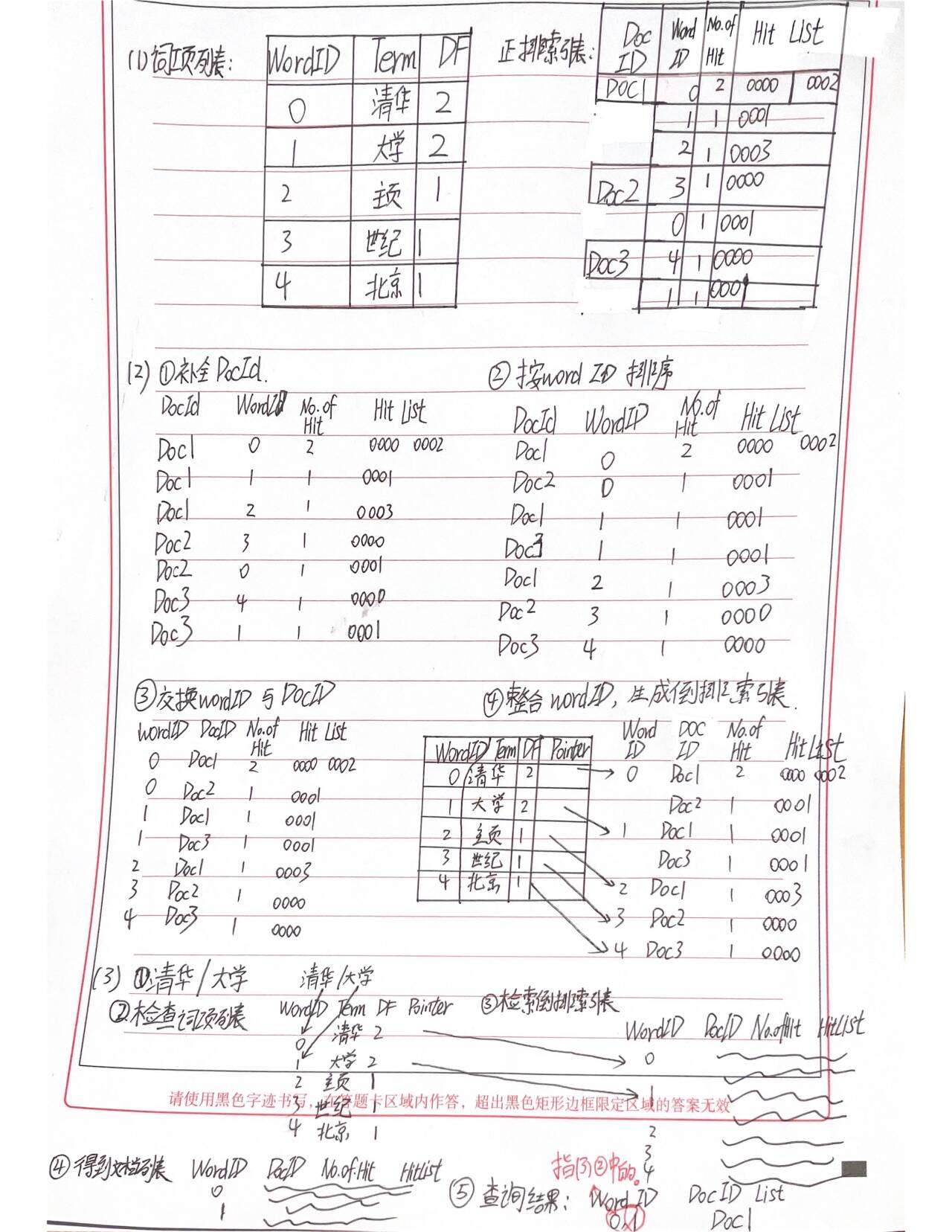


## 倒排索引

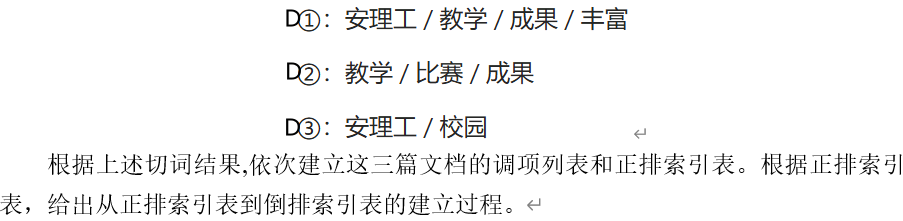
### 例题1



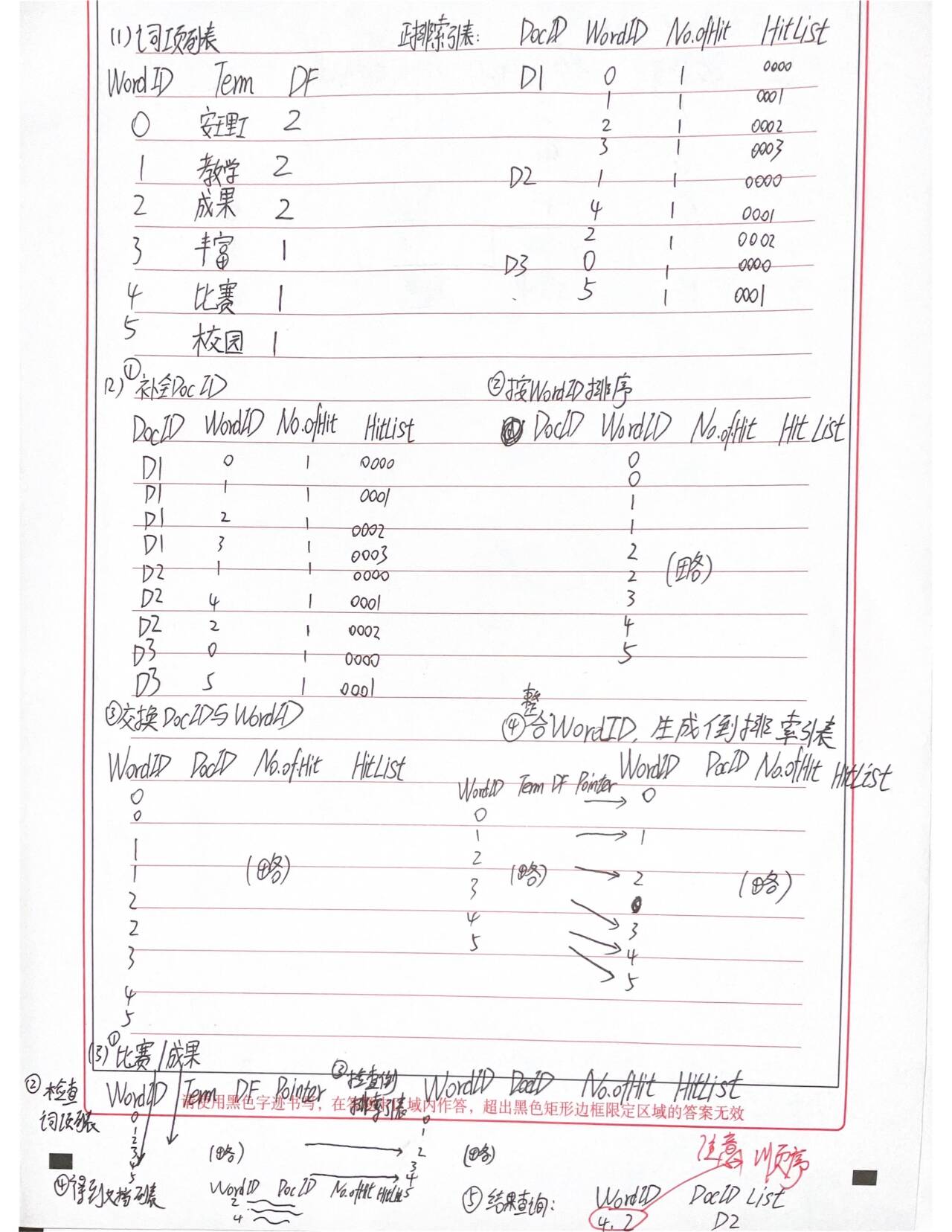
根据上述切词结果,依次建立这三篇文档的词项列表和正排索引表。根据正排索引表，给出从正排索引表到倒排索引表的建立过程。查询：清华大学。



### 例题2



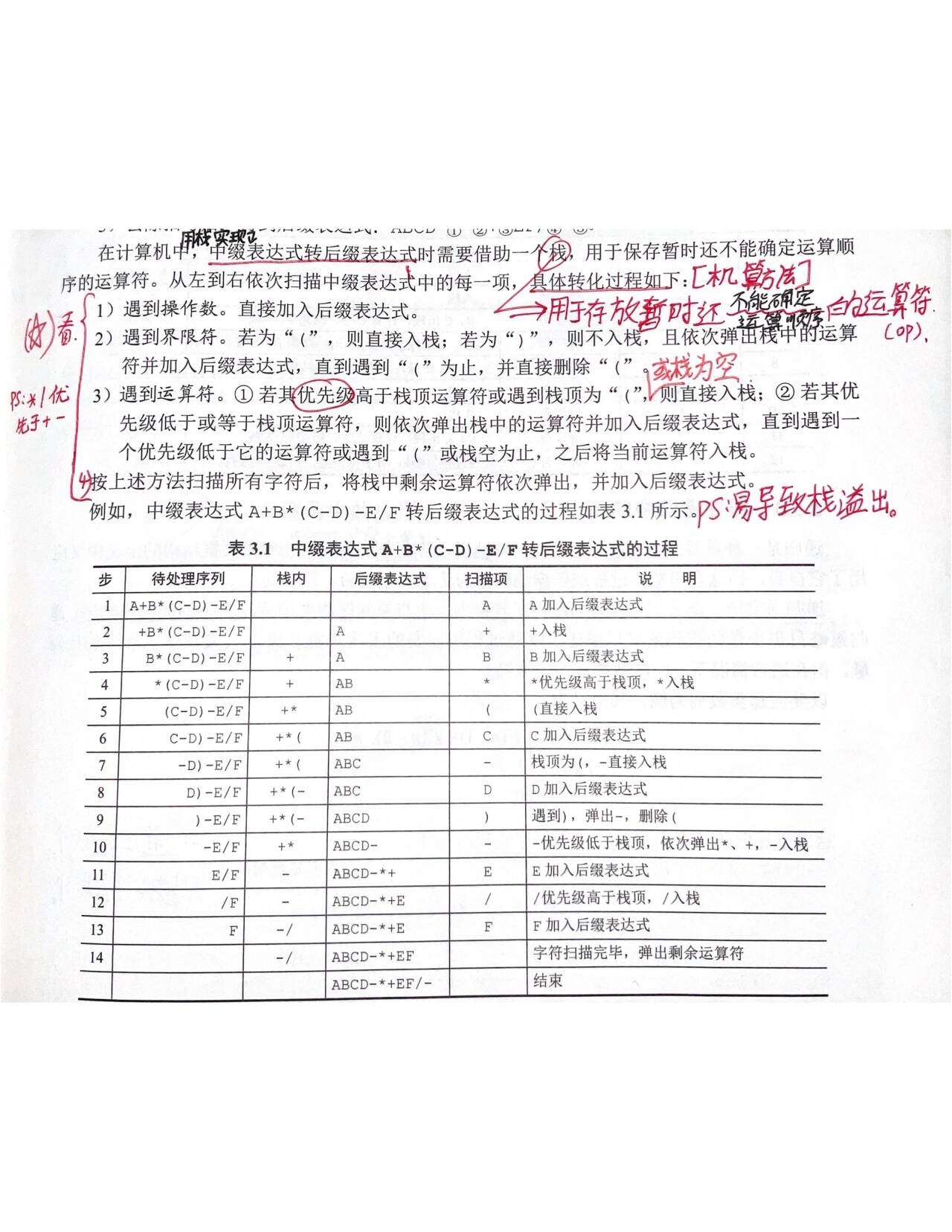
根据上述切词结果,依次建立这三篇文档的词项列表和正排索引表。根据正排索引表，给出从正排索引表到倒排索引表的建立过程。查询：比赛成果。



## 逆波兰表达式

### 规则

Ps：后缀表达式也叫“逆波兰表达式”。



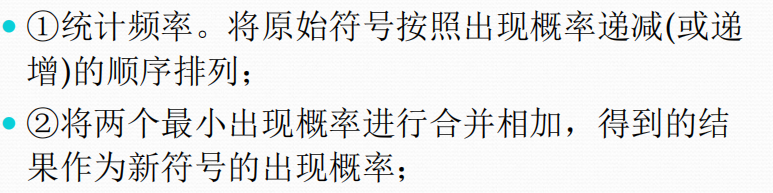
### 例题

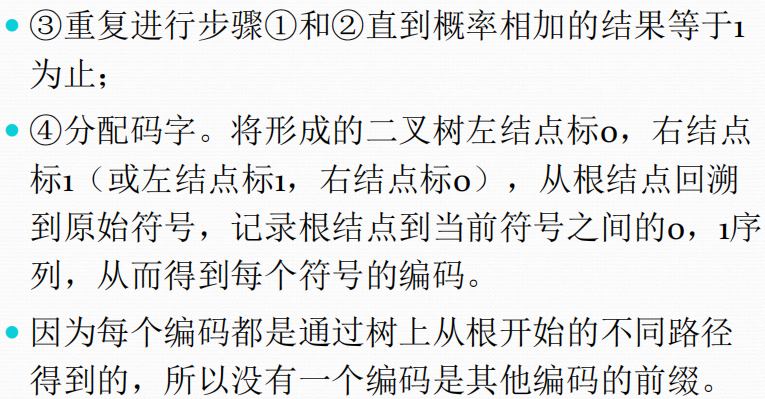


## 文本压缩技术

### 哈夫曼编码

规则





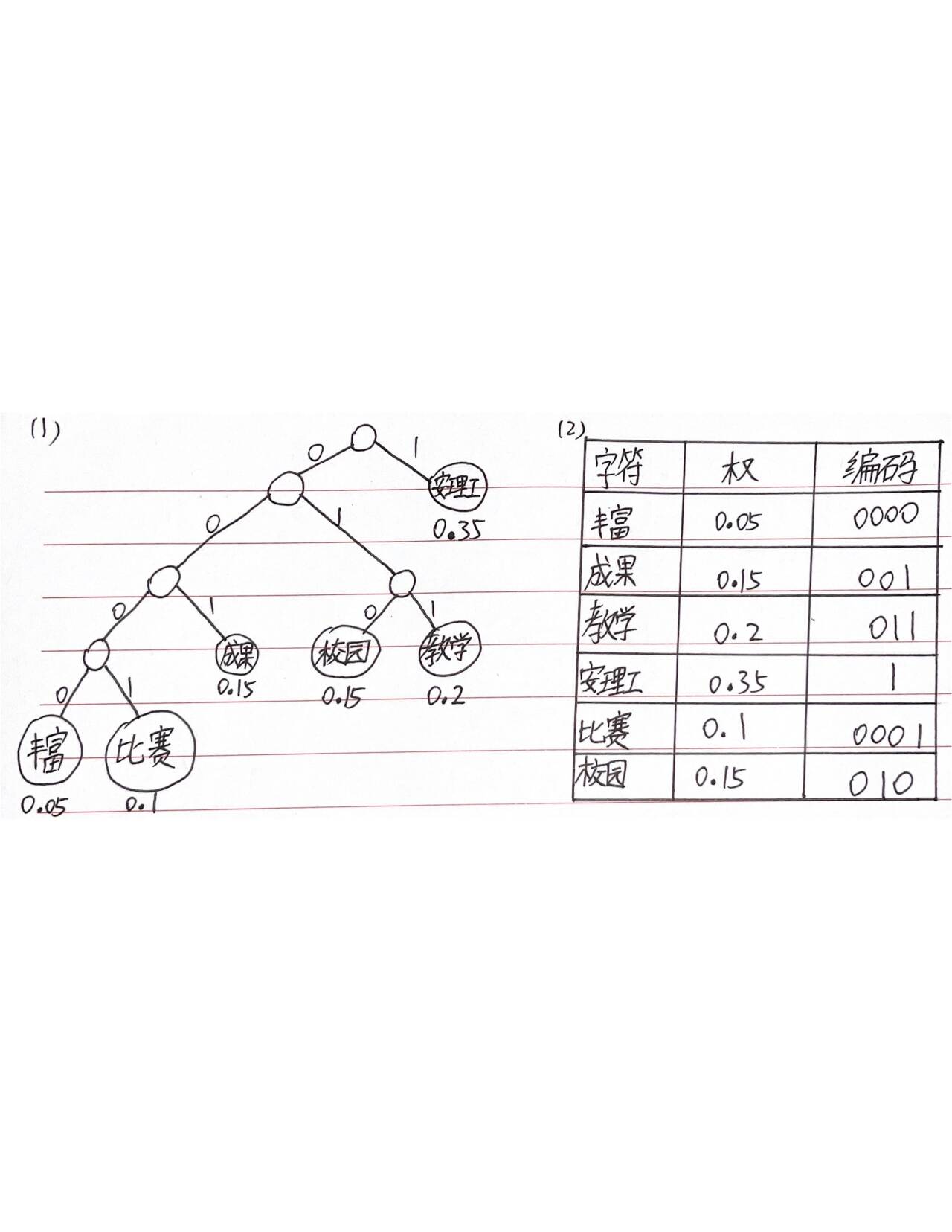
Ps：哈夫曼树不唯一。

例题

在整个索引文档库中索引词出现的概率如下表。利用霍夫曼树对索引词进行编码，画出霍夫曼树。根据霍夫曼编码，进行编码分配，写出最后的编码分配表。

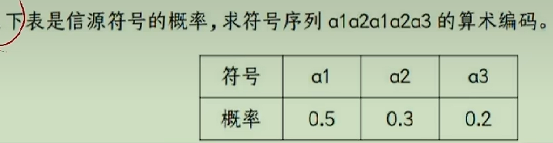
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 索引词 | 丰富 | 成果 | 教学 | 安理工 | 比赛 | 校园 |
| 出现的概率 | 0.05 | 0.15 | 0.2 | 0.35 | 0.1 | 0.15 |

解析

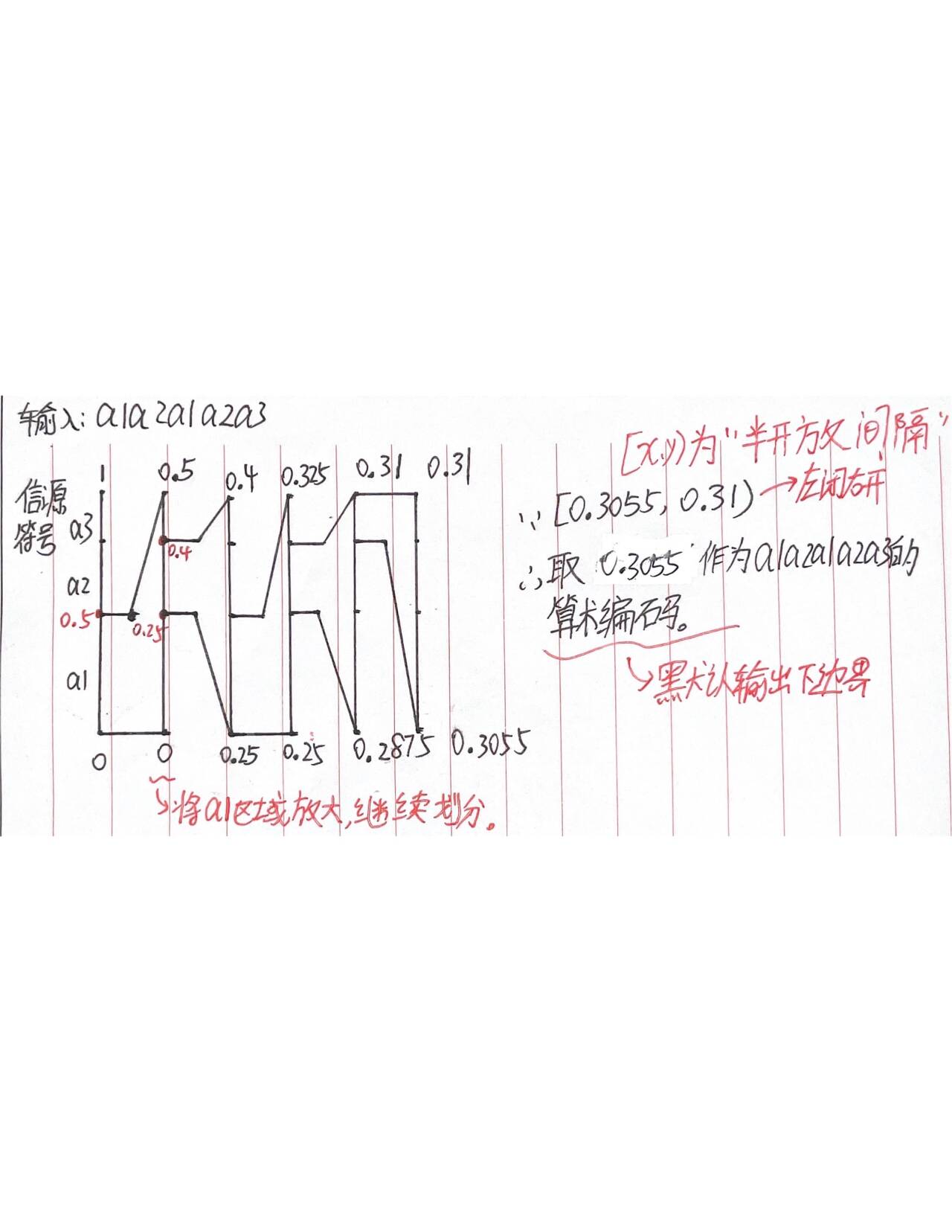


### 算术编码

例题1



解析1

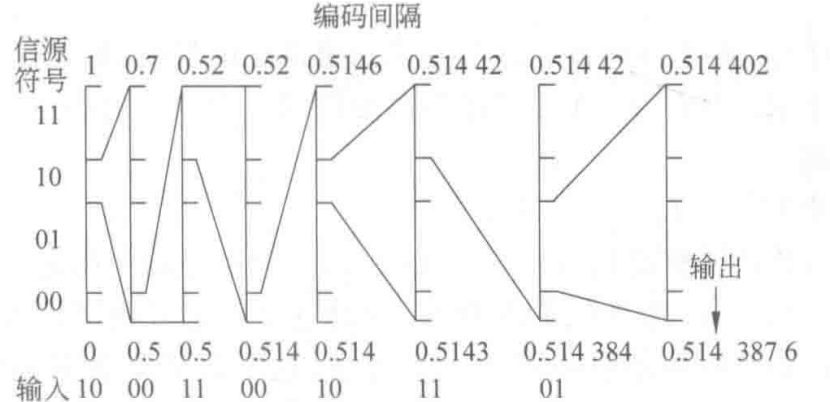


例题2

假设信源符号为{00,01,10,11}，这些符号的概率分别为{0.1,0.4,0.2,0.3}，

求10 00 11 00 10 11 01的算术编码。

解析2

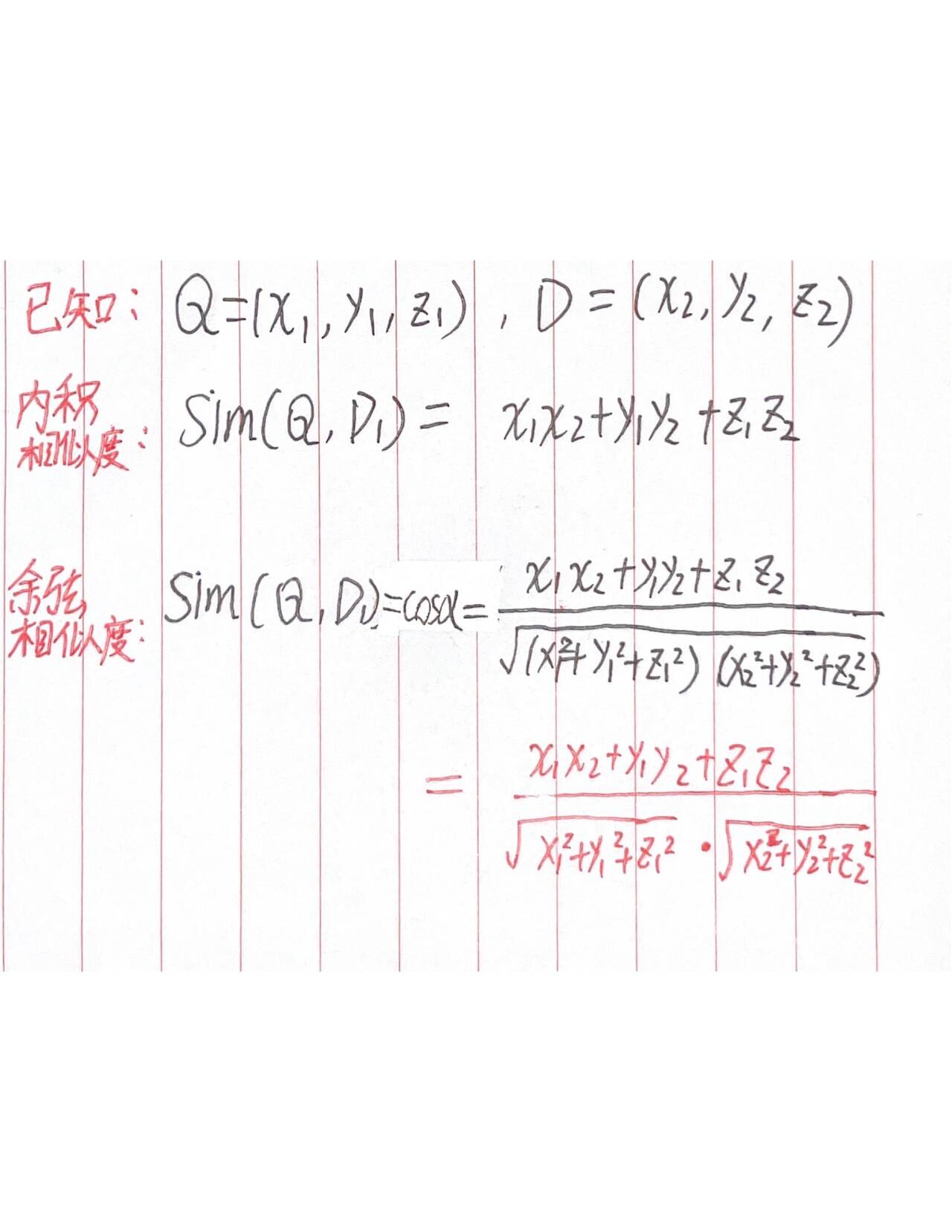


# Chapt7 信息查询与评价技术

## 向量空间模型

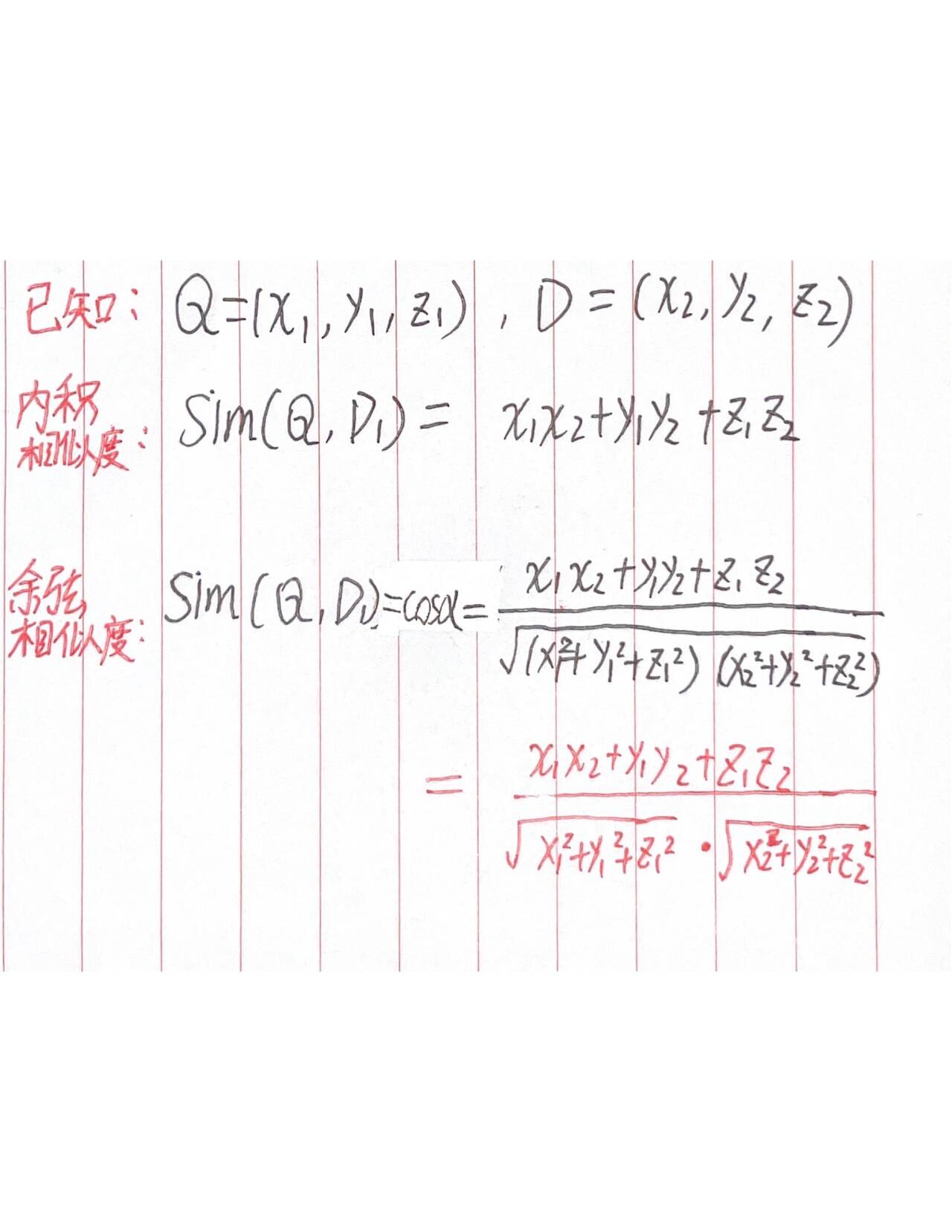
### 内积相似度

内积相似度越大，相关度越高。



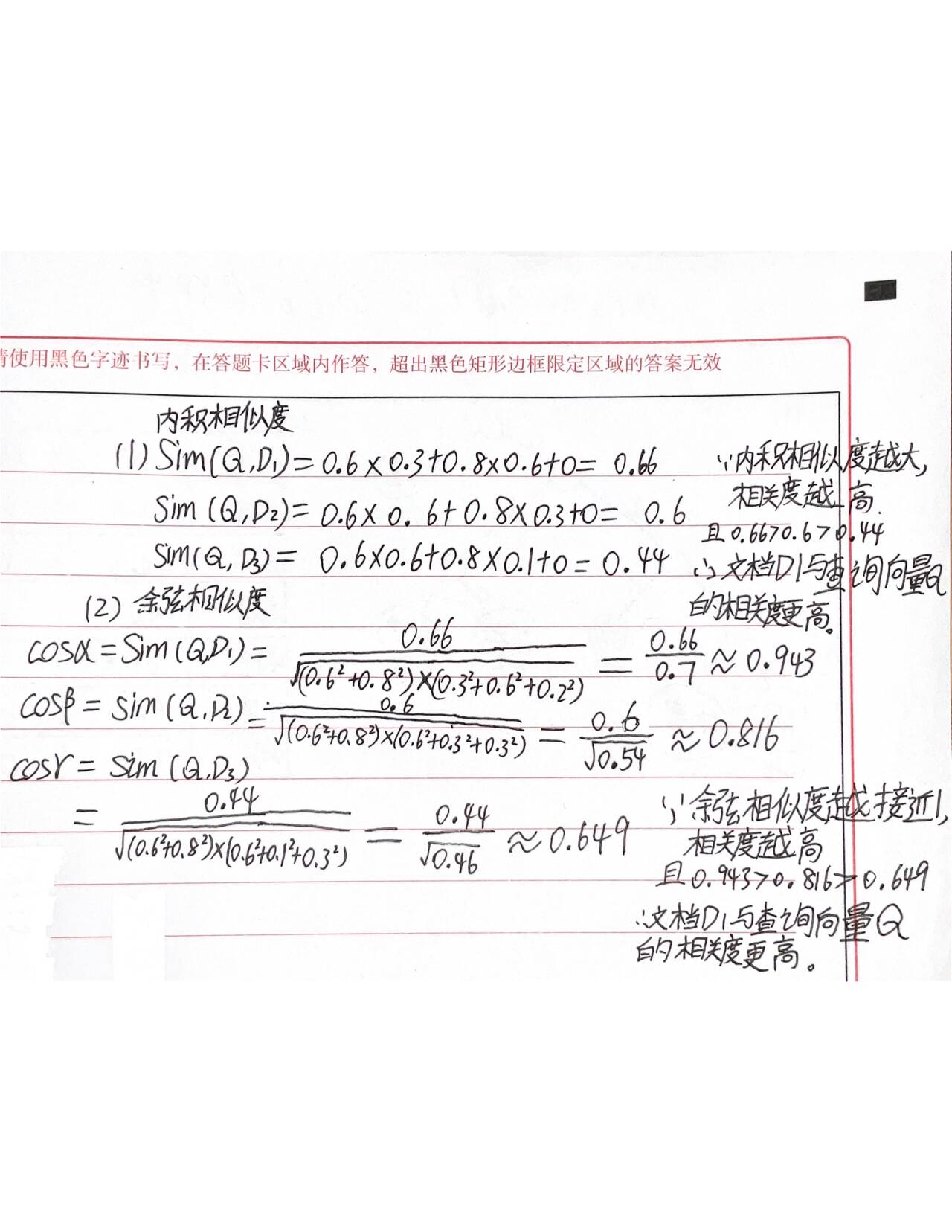
### 余弦相似度

余弦相似度越接近1，相关度越高。



### 例题

查询向量Q=(0.6,0.8,0)，三个文档D1、D2和D3的文档向量为D1=(0.3,0.6,0.2)、D2=(0.6,0.3,0.3)和D3=(0.6,0.1,0.3)。（1）请分别计算查询向量与三个文档的内积相似度，并结合该相似度说明哪个文档与查询相关度更高？（2）请分别计算查询向量与三个文档的余弦相似度，并结合该相似度说明哪个文档与查询相关度更高？



### 分析二者的优劣

|  |  |  |
| --- | --- | --- |
|  | 内积相似度 | 余弦相似度 |
| 优点 | 计算简单 | 对内积相似度进行了归一化处理，避免了文档长度对相似度度量的影响。 |
| 缺点 | 文档越长，文档对应的向量权重就会越大。内积相似度会在较长文档上得到较大相似度。 | 在一些需要考虑向量长度信息的应用场景中表现不佳。 |

## 万维网链接结构图

## HITS算法

## PageRank算法

## 搜索引擎评价指标