《信息检索导论》课后练习答案

王斌

最后更新日期　2013/9/28

第一章布尔检索

习题 **1-1 [*]**　画出下列文档集所对应的倒排索引（参考图　　1-3 中的例子）。

文档　1 new home sales top forecasts

文档　2 home sales rise in july

文档　3 increase in home sales in july

文档　4 july new home sales rise

解答：

forecasts	------->	1			
home	------->	1	2	3	4
in	------->	2	3		
increase	------->	3			
july	------->	2	3	4	
new	------->	1	4		
rise	------->	2	4		
sales	------->	1	2	3	4
top	------->	1			

习题 **1-2** [*]　考虑如下几篇文档：

文档 1 breakthrough drug for schizophrenia

文档 2 new schizophrenia drug

文档 3 new approach for treatment of schizophrenia

文档 4 new hopes for schizophrenia patients

a. 画出文档集对应的词项—文档矩阵；

解答：

	文档 **1**	文档 **2**	文档 **3**	文档 **4**
approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
for	1	0	1	1

hopes	0	0	0	1
new	0	1	1	1
of	0	0	1	0
patients	0	0	0	1
schizophrenia	1	1	1	1
treatment	0	0	1	0

b. 画出该文档集的倒排索引（参考图 1-3 中的例子）。

解答：参考 a。

习题 **1-3 [*]** 对于习题 1-2 中的文档集，如果给定如下查询，那么返回的结果是什么？

a. schizophrenia AND drug

解答：{文档 1，文档 2}

b. for ANDNOT (drug OR approach)

解答：{文档 4}

习题 **1-4** [*] 对于如下查询，能否仍然在 O(x+y)次内完成？其中 x 和 y 分别是 Brutus 和 Caesar所对应的倒排记录表长度。如果不能的话，那么我们能达到的时间复杂度是多少？

a. Brutus AND NOT Caesar

b. Brutus OR NOT Caesar

解答：

a. 可以在 O(x+y)次内完成。通过集合的减操作即可。具体做法参考习题 1-11。

b. 不能。不可以在 O(x+y)次内完成。 因为 NOT Caesar的倒排记录表需要提取其他所有词项对应的倒排记录表。所以需要遍历几乎全体倒排记录表，于是时间复杂度即为所有倒排记录表的长度的和 N，即 O(N) 或者说 O(x+N-y)。

习题 **1-5 [*]** 将倒排记录表合并算法推广到任意布尔查询表达式，其时间复杂度是多少？比如，对于查询

c. (Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

我们能在线性时间内完成合并吗？这里的线性是针对什么来说的？我们还能对此加以改进吗？

解答：时间复杂度为 O(qN)，其中 q 为表达式中词项的个数， N 为所有倒排记录表长度之和。 也就是说可以在词项个数 q 及所有倒排记录表长度 N 的线性时间内完成合并。由于任意布尔表达式处理算法复杂度的上界为 O(N)，所以上述复杂度无法进一步改进。

习题 **1-6 [**]** 假定我们使用分配律来改写有关 AND 和 OR 的查询表达式。

a. 通过分配律将习题 1-5 中的查询写成析取范式；

b. 改写之后的查询的处理过程比原始查询处理过程的效率高还是低？

c. 上述结果对任何查询通用还是依赖于文档集的内容和词本身？

解答：

a. 析取范式为： (Brutus And Not Anthony And Not Cleopatra) OR (Caesar ANDNOTAnthony ANDNOTCleopatra)

b. 这里的析取范式处理比前面的合取范式更有效。这是因为这里先进行 AND 操作 (括号内)，得到的倒排记录表都不大，再进行 OR操作效率就不会很低。而前面需要先进行 OR操作，得到的中间倒排记录表会更大一些。

c. 上述结果不一定对，比如两个罕见词 A 和 B构成的查询 (A OR B) AND NOT(HONG OR KONG) ，假设 HONG KONG 一起出现很频繁。此时合取方式可能处理起来更高效。如果在析取范式中仅有词项的非操作时， b中结果不对。

习题 **1-7 [*]** 请推荐如下查询的处理次序。

d. (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)

其中，每个词项对应的倒排记录表的长度分别如下：

词项	倒排记录表长度
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

解答：

由于：

(tangerine OR trees) 46653+316812 = 363465
(marmalade OR skies) 107913+271658 = 379571
(kaleidoscope OR eyes) 87009+213312 = 30321

所以推荐处理次序为：

(kaleidoscope OR eyes) AND (tangerine OR trees) AND (marmalade OR skies)

习题 **1-8[*]** 对于查询

e. friends AND romans AND (NOT countrymen)

如何利用 countrymen 的文档频率来估计最佳的查询处理次序？特别地， 提出一种在确定查询顺序时对逻辑非进行处理的方法。

解答：令 friends、romans 和 countrymen 的文档频率分别为 x、y、z。如果 z 极高，则将 N-z 作为 NOT countrymen 的长度估计值，然后按照 x、 y、 N-z 从小到大合并。如果 z 极低，则按照 x、 y、 z 从小到大合并。

习题 **1-9 [**]** 对于逻辑与构成的查询，按照倒排记录表从小到大的处理次序是不是一定是最优的？如果是，请给出解释；如果不是，请给出反例。

解答：不一定。比如三个长度分别为 x,y,z 的倒排记录表进行合并，其中 x>y>z，如果 x 和 y 的交集为空集，那么有可能先合并 x、y 效率更高。

习题 **1-10** [**] 对于查询 xORy，按照图 1-6 的方式，给出一个合并算法。

解答：

```
1   answer<- ( )
2   while p1!=NIL and p2!=NIL
3   do if docID(p1)=docID(p2)
4   then    ADD(answer,docID(p1))
5           p1<- next(p1)
6     p2<-next(p2)
```

```
7       else if docID(p1)<docID(p2)
8       then    ADD(answer,docID(p1))
9p1<- next(p1)
10else          ADD(answer,docID(p2))
11              p2<-next(p2)
12if p1!=NIL        // x 还有剩余
13    then while p1!=NIL do ADD (answer, docID(p1))
14    else while p2!=NIL do ADD(answer,docID(p2))
15 return(answer)
```

习题 1-11 [*] 如何处理查询 x AND NOT y ？为什么原始的处理方法非常耗时？给出一个针对该查询的高效合并算法。

解答：由于 NOT y几乎要遍历所有倒排表，因此如果采用列举倒排表的方式非常耗时。可以采用两个有序集合求减的方式处理 x AND NOT y。算法如下：

```
Meger(p1,p2)
1       answer ⟵ ()
2       while p1!=NIL and p2!=NIL
3       do if docID(p1) =docID(p2)
4       then  p1    next(p1)
5             p2    next(p2)
6       else if docID(p1)<docID(p2)
7       then  ADD(answer, docID(p1))
8             p1    next(p1)
9       else  ADD(answer, docID(p2))
10            p2    next(p2)
11      if p1!=NIL        // x 还有剩余
12      then while p1!=NIL do ADD (answer, docID(p1))
13      return(answer)
```

习题 **1-12** [*] 利用 Westlaw系统的语法构造一个查询， 通过它可以找到 professor、teacher或 lecturer 中的任意一个词，并且该词和动词 explain 在一个句子中出现，其中 explain 以某种形式出现。

解答： professor teacher lecturer /s explain!

习题 **1-13** [*] 在一些商用搜索引擎上试用布尔查询，比如，选择一个词（如 burglar），然后将如下查询提交给搜索引擎

(i) burglar ；(ii)burglar AND burglar ；(iii) burglar OR burglar。

对照搜索引擎返回的总数和排名靠前的文档, 这些结果是否满足布尔逻辑的意义？对于大多数搜索引擎来说， 它们往往不满足。 你明白这是为什么吗？如果采用其他词语, 结论又如何？比如以下查询

(i) knight ；(ii) conquer ；(iii) knight OR conquer 。

第二章词汇表和倒排记录表

习题 **2-1** [*] 请判断如下说法是否正确。

a. 在布尔检索系统中，进行词干还原从不降低正确率。

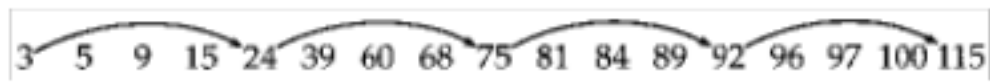
b. 在布尔检索系统中，进行词干还原从不降低召回率。

c. 词干还原会增加词项词典的大小。

d. 词干还原应该在构建索引时调用，而不应在查询处理时调用。

解答：a 错 b 对 c 错 d 错

习题 **2-7** [*] 考虑利用如下带有跳表指针的倒排记录表



和一个中间结果表（如下所示，不存在跳表指针）进行合并操作。

3 5 89 95 97 99 100 101

采用图 2-10 所示的倒排记录表合并算法，请问：

a. 跳表指针实际跳转的次数是多少（也就是说，指针 p_1 的下一步将跳到 skip （p1））？

一次， 24—>75

b. 当两个表进行合并时，倒排记录之间的比较次数是多少？【如下答案不一定正确，有人利用程序计算需要 21 次，需要回到算法，本小题不扣分，下面不考虑重新比较同意对数字】

解答： 18 次： <3,3>, <5,5>, <9,89>, <15,89>,<24,89>,<75,89>,<92,89>,<81,89>,<84,89>,<89,89>,<92,95>,<115,95>,<96,95>,<96,97>,<97,97>,<100,99>,<100,100><115,101>

c. 如果不使用跳表指针，那么倒排记录之间的比较次数是多少？

解答： 19 次： <3,3>,<5,5>,<9,89>,<15,89>,<24,89>,<39,89>,<60,89>,<68,89>,<75,89>,<81,89>,<84,89>,<89,89><92,95>, <96,95>,<96,97>,<97,97>,<100,99>,<100,100>,<115,101>

习题 **2-9** [*] 下面给出的是一个位置索引的一部分，格式为：词项 ：文档 1: 〈位置 1，位置 2,…〉;文档 2: 〈位置 1，位置 2,…〉。

angels: 2: 〈36,174,252,651〉; 4: 〈12,22,102,432〉; 7: 〈17〉;

fools: 2: 〈1,17,74,222〉; 4: 〈8,78,108,458〉; 7: 〈3,13,23,193〉;

fear: 2: 〈87,704,722,901〉; 4: 〈13,43,113,433〉; 7: 〈18,328,528〉;

in: 2: 〈3,37,76,444,851〉; 4: 〈10,20,110,470,500〉; 7: 〈5,15,25,195〉;

rush: 2: 〈2,66,194,321,702〉; 4: 〈9,69,149,429,569〉; 7: 〈4,14,404〉;

to: 2: 〈47,86,234,999〉; 4: 〈14,24,774,944〉; 7: 〈199,319,599,709〉;

tread: 2: 〈57,94,333〉; 4: 〈15,35,155〉; 7: 〈20,320〉;

where: 2: 〈67,124,393,1001〉; 4: 〈11,41,101,421,431〉; 7: 〈16,36,736〉;

那么哪些文档和以下的查询匹配？其中引号内的每个表达式都是一个短语查询。

a. “ fools rush in ”

解答：文档 2、4、7

b. “ fools rush in ” AND “ angels fear to tread ”

解答：文档 4

第三章词典及容错式检索

习题 **3-5** 再次考虑 3.2.1 节中的查询 fi*mo*er ，如果采用 2-gram 索引的话，那么对应该查询应该会产生什么样的布尔查询？你能否举一个词项的例子，使该词匹配 3.2.1 节的轮排索引查询，但是并不满足刚才产生的布尔查询？

解答： 2-gram 索引下的布尔查询： $f AND fi AND mo AND er AND r$

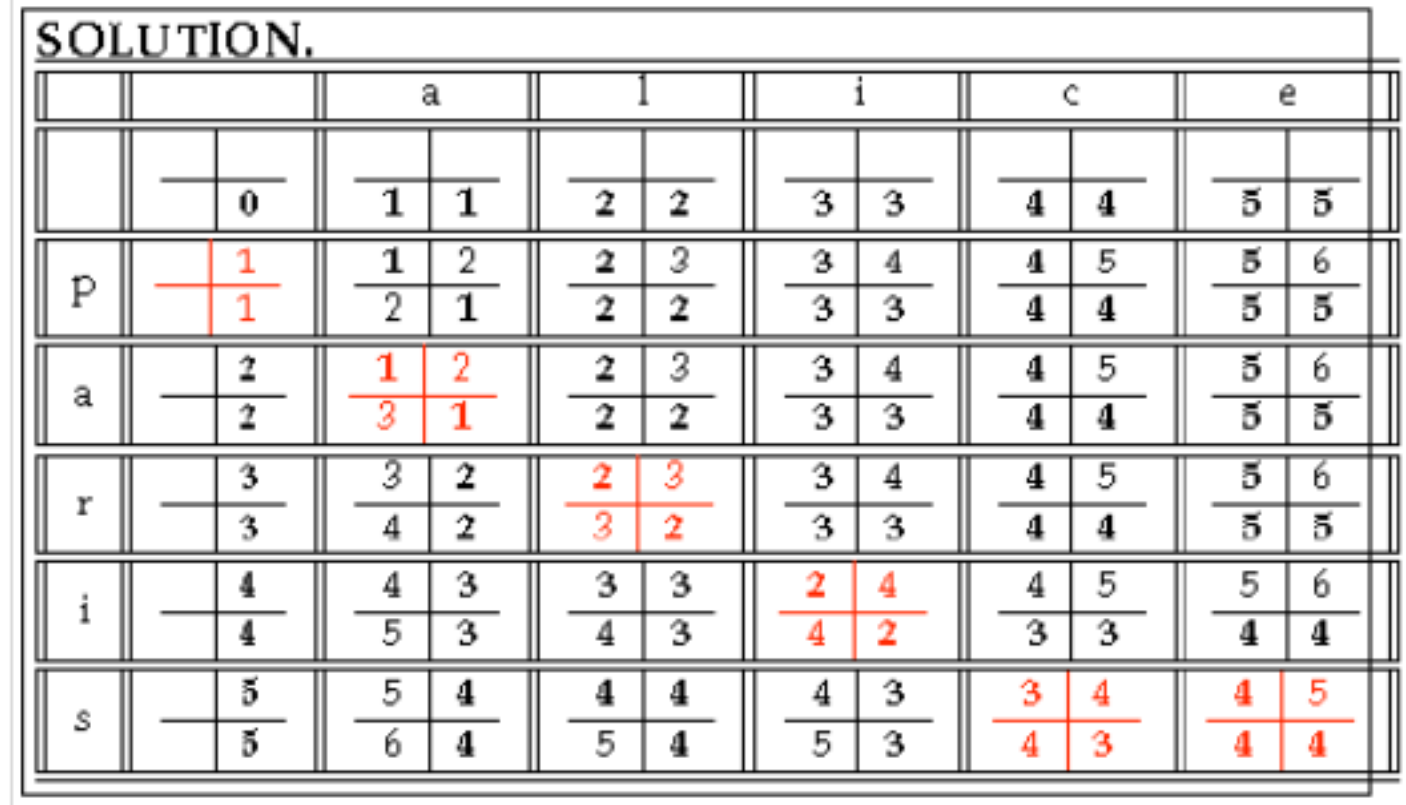
词项 filibuster(海盗)满足 3.2.1 节的轮排索引查询，但是并不满足上述布尔查询

习题 3-7 如果 |si | 表示字符串 si 的长度，请证明 s1 和 s2 的编辑距离不可能超过 max{|s1|, |s2|} 。

证明：不失一般性，假设 |s1|<= |s2| ，将 s1 转换为 s2 的一种做法为：将 s1 中的每个字符依次替换为 s2 中的前 |s1| 个字符，然后添加 s2 的后 |s2|-|s1| 个字符，上述操作的总次数为 |s2|= max{|s1|, |s2|} ，根据编辑距离的定义，其应该小于 |s2|= max{|s1|, |s2|}

习题 **3-8** 计算 paris 和 alice 之间的编辑距离，给出类似于图 3-5 中的算法结果，其中的 5 × 5 矩阵包含每个前缀子串之间的计算结果。

解答：



习题 **3-11** 考虑四词查询 catched in the rye ，假定根据独立的词项拼写校正方法，每个词都有 5 个可选的正确拼写形式。 那么， 如果不对空间进行缩减的话， 需要考虑多少可能的短语拼写形式（提示：同时要考虑原始查询本身，也就是每个词项有 6 种变化可能）？

解答： 6*6*6*6=1296

习题 **3-14** 找出两个拼写不一致但 soundex 编码一致的专有名词。

解答：Mary, Mira (soundex 相同)，本题答案不唯一， 可能有其他答案， 但是 soundex 编码必须一致。

第四章索引构建

? 习题 **4-1** 如果需要 T $\log_2 T$ 次比较（ T 是词项 ID—文档 ID 对的数目），每次比较都有两次磁盘寻道过程。假定使用磁盘而不是内存进行存储，并且不采用优化的排序算法（也就是说不使用前面提到的外部排序算法），那么对于 Reuters-RCV1构建索引需要多长时间？计算时假定采用表 4-1 中的系统参数。

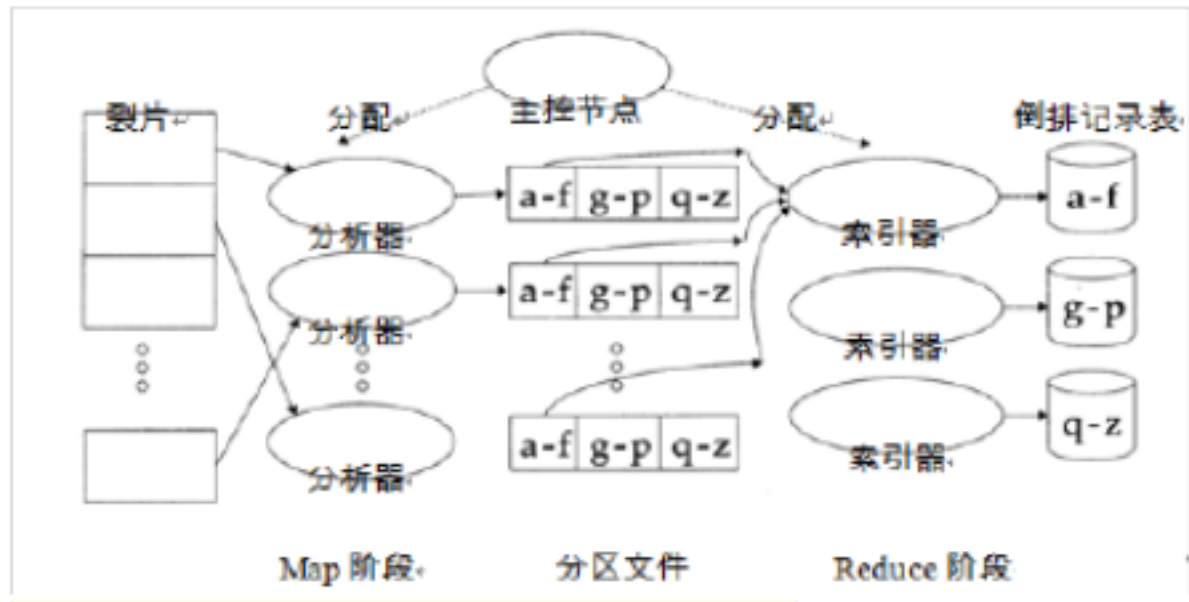
解答：

对于 Reuters-RCV1，$T=10^8$

因此排序时间（文档分析时间可以忽略不计）为：$2*(10^8*\log_2 10^8)*5*10^{-3}s = 26575424s = 7382\ h=308\ day$

习题 **4-3** 对于 n = 15 个数据片，r = 10 个分区文件，j = 3 个词项分区，假定使用的集群的机器的参数如表 4-1 所示，那么在 MapReduce 构架下对 Reuters-RCV1语料进行分布式索引需要多长时间？

【给助教：教材不同印刷版本表 4-2 不一样，不同同学用的不同版本，还有本题过程具有争议。暂不扣分】



解答【整个计算过程是近似的，要了解过程】：

(一)、MAP 阶段【读入语料（已经不带 XML 标记信息了，参考表 5-6），词条化，写入分区文件】：

(1) 读入语料：

基于表 4-2，Reuters RCV1共有 $8*10^5$ 篇文档，每篇文档有 200 词条，每个词条（考虑标点和空格）占 6B，因此整个语料库的大小为 $8*10^5*200*6=9.6*10^8$B （近似 1GB，注表 4-2 对应于表 5-1 第 3 行的数据，而那里的数据已经经过去数字处理，因此实际的原始文档集大小应该略高于 0.96G，这里近似计算，但是不要认为没有处理就得到表 5-1 第 3 行的结果）

将整个语料库分成 15 份，则每份大小为 $9.6*10^8/15$ B

每一份读入机器的时间为：$9.6*10^8/15*2*10^{-8}=1.28s$

(2) 词条化：每一份语料在机器上进行词条化处理，得到 $8*10^5*200=1.6*10^8$ 个词项 ID-文档 ID 对(参考表 4-2 和图 4-6，注意此时重复的词项 ID-文档 ID 对还没有处理），共占 $1.6*10^8*8=1.28*10^9$ 个字节，词条化的时间暂时忽略不计【从题目无法得到词条化这一部分时间，从表 5-1 看词条化主要是做了去数字和大小写转换，当然也感觉这一部分的处理比较简单，可以忽略】。

(3) 写入分区文件：每一份语料得到的词项 ID-文档 ID (Key-Value)存储到分区所花的时间为：$(1.28*10^9/15)*2*10^{-8}=1.71s$

(4) MAP 阶段时间：

由于分成 15 份，但只有 10 台机器进行 MAP 操作，所以上述 MAP 操作需要两步，因此，整个 MAP

过程所需时间为 (1.28+1.71)*2=6.0s

(二)、REDUCE阶段【读入分区文件，排序，写入倒排索引】：

(1) 读入分区文件【读入过程中已经实现所有 Key-Value对中的 Value按 Key聚合，即变成 Key, list(V1,V2..)。聚合过程在内存中实现，速度很快，该时间不计。另外，网络传输时间这里也不计算】：

根据表 4-2,所有倒排记录的数目为 $1.6*10^8$, 因此 3台索引器上每台所分配的倒排记录数目为 $1.6*10^8/3$, 而每条记录由 4 字节词项 ID 和 4 字节文档 ID 组成，因此每台索引器上需要读入的倒排记录表数据为 $1.28*10^9/3$ 字节。

于是，每台索引器读数据的时间为 $1.28*10^9/3*2*10^{-8}$=8.5s

(2) 排序：

每台索引器排序所花的时间为 $1.6*10^8/3*\log_2(1.6*10^8/3)*10^{-8}$=13.7s

(3) 写入倒排索引文件【此时倒排文件已经实现文档 ID 的去重，假定只存储词项 ID 和文档 ID 列表，并不存储其他信息 (如词项的 DF及在每篇文档中的 TF还有指针等等)】：

需要写入磁盘的索引大小为 (据表 4-2，词项总数为 $4*10^5$个) $4*10^5/3*4+10^8/3*4=4/3*10^8$字节

索引写入磁盘的时间为： $4/3*10^8*2*10^{-8}$=2.7s

(4) REDUCE阶段时间为： 8.5+13.7+2.7=24.9

(三) 因此，整个分布式索引的时间约为 6.0+8.5+13.7+2.7=30.9s

第五章索引压缩

习题 **5-2** 估计 Reuters-RCV1文档集词典在两种不同按块存储压缩方法下的空间大小。其中，第一种方法中 k = 8，第二种方法中 k = 16。

解答：

每 8 个词项会节省 7*3 个字节，同时增加 8 个字节，于是每 8 个词项节省 7*3-8=13 字节，所有词项共节省 13*400000/8=650K，因此，此时索引大小为 7.6MB-0.65MB=6.95MB

每 16 个词项会节省 15*3 个字节，同时增加 16 个字节，于是每 16 个词项节省 15*3-16=29 字节，所有词项共节省 29*400000/16=725K，因此，此时索引大小为 7.6MB-0.725MB=6. 875MB

习题 **5-6** 考虑倒排记录表（4, 10, 11, 12, 15, 62, 63, 265, 268, 270, 400）及其对应的间距表（4, 6, 1, 1, 3, 47, 1, 202,3, 2, 130)。假定倒排记录表的长度和倒排记录表分开独立存储，这样系统能够知道倒排记录表什么时候结束。采用可变字节码：

(i) 能够使用 1 字节来编码的最大间距是多少？

(ii) 能够使用 2 字节来编码的最大间距是多少？

(iii) 采用可变字节编码时， 上述倒排记录表总共需要多少空间 （只计算对这些数字序列进行编码的空间消耗）？

解答：

(i) 2^7-1=127 (答 128 也算对，因为不存在 0 间距， 0 即可表示间距 1，……)

(ii) $2^{14}-1$=16383 (答 16384 也算对)

(iii) 1+1+1+1+1+1+1+2+1+1+2=13

习题 **5-8 [*]** 对于下列采用 γ 编码的间距编码结果，请还原原始的间距序列及倒排记录表。

11100011101010111111011011110 11

解答：

1110 001; 110 10; 10 1; 111110 11011; 110 11

1001; 110; 11; 111011; 111

9; 6; 3; 32+16+8+2+1=59; 7

9; 15;18;77;84

第六章文档评分、词项权重计算及向量空间模型

习题 **6-10** 考虑图 6-9 中的 3 篇文档 Doc1、Doc2、Doc3 中几个词项的 tf 情况，采用图 6-8 中的 idf 值来计算所有词项 car、auto、insurance 及 best 的 tf-idf 值。

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

图 6-9 习题 6-10 中所使用的 tf 值

解答：

$idf_{car}=1.65$， $idf_{auto}=2.08$， $idf_{insurance}=1.62$， $idf_{best}=1.5$，

于是，各词项在各文档中的 tf-idf 结果如下表：

	Doc1	Doc2	Doc3
car	27*1.65=44.55	4*1.65=6.6	24*1.65=39.6
auto	3*2.08=6.24	33*2.08=68.64	0
insurance	0	33*1.62=53.46	29*1.62=46.98
best	14*1.5=21	0	17*1.5=25.5

习题 **6-12** 公式（6-7）中对数的底对公式（ 6-9）会有什么影响？对于给定查询来说，对数的底是否会对文档的排序造成影响？

解答：没有影响。

假定 idf 采用与 (6-7)不同的底 x 计算，根据对数换底公式有。

$idf_t(x)=\log_x(N/df_t)=\log(N/df_t)/\log x=idf_t/\log x$，

由于 idft(x) 和 idft 之间只相差一个常数因子 1/logx，在公式 (6-9) 的计算中该常数可以作为公因子提出，因此文档的排序不会改变。

习题 **6-19** 计算查询 digital cameras 及文档 digital cameras and video cameras 的向量空间相似度并将结果填入表 6-1 的空列中。假定 N=10000000，对查询及文档中的词项权重（ wf 对应的列）采用对数方法计算，查询的权重计算采用 idf，而文档归一化采用余弦相似度计算。将 and 看成是停用词。请在 tf 列中给出词项的出现频率，并计算出最后的相似度结果。

表 6-1 习题 6-19 中的余弦相似度计算

词	查询					文档			q_i d_i
	tf	wf	df	idf	q_i=wf-idf	tf	wf	d_i=归一化的 wf	
digital			10 000						
video			100 000						
cameras			50 000						

解答：【本质上这里没有考虑查询向量的归一化，即没有考虑查询向量的大小，严格上不是余弦相似度】

词	查询					文档			q_i d_i
	tf	wf	df	idf	q_i=wf-idf	tf	wf	d_i=归一化的 wf	
digital	1	1	10 000	3	3	1	1	0.520	
video	0	0	100 000	2	0	1	1	0.520	3.112
cameras	1	1	50 000	2.301	2.301	2	1.301	0.677	

习题 **6-23** 考虑习题 6-10 中 4 个词项和 3 篇文档中的 tf 和 idf 值，采用如下权重计算机制来计算获得得分最高的两篇文档： (i) nnn.atc ；(ii) ntc.atc。

解答： (i) 根据题意文档采用 nnn，查询采用 atc，然后计算内积，于是有：

词项	查询 q				文档 Doc1			得分
	tf	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	
car	1	1.65	1.65	0.560	27	1	27	23.310
auto	0.5	2.08	1.04	0.353	3	1	3	
insurance	1	1.62	1.62	0.550	0	1	0	
best	1	1.5	1.5	0.509	14	1	14	

词项	查询 q				文档 Doc2			得分
	tf	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	
car	1	1.65	1.65	0.560	4	1	4	32.037
auto	0.5	2.08	1.04	0.353	33	1	33	
insurance	1	1.62	1.62	0.550	33	1	33	
best	1	1.5	1.5	0.509	0	1	0	

词项	查询 q				文档 Doc3			得分
	tf	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	
car	1	1.65	1.65	0.560	24	1	24	38.046
auto	0.5	2.08	1.04	0.353	0	1	0	
insurance	1	1.62	1.62	0.550	29	1	29	
best	1	1.5	1.5	0.509	17	1	17	

于是，在 nnn.atc 下， Score(q,Doc3)>Score(q,Doc2)>Score(q,Doc1)

(ii) 根据题意文档采用 ntc，查询采用 atc，然后计算内积，于是有：

词项	查询 q				文档 Doc1				得分
	tf(a)	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	归一化 tf-idf	
car	1	1.65	1.65	0.560	27	1.65	44.55	0.897	0.76
auto	0.5	2.08	1.04	0.353	3	2.08	6.24	0.125	
insurance	1	1.62	1.62	0.550	0	1.62	0	0	
best	1	1.5	1.5	0.509	14	1.5	21	0.423	

词项	查询 q				文档 Doc2				得分
	tf(a)	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	归一化 tf-idf	
car	1	1.65	1.65	0.560	4	1.65	6.6	0.075	0.66
auto	0.5	2.08	1.04	0.353	33	2.08	68.64	0.786	
insurance	1	1.62	1.62	0.550	33	1.62	53.46	0.613	
best	1	1.5	1.5	0.509	0	1.5	0	0	

词项	查询 q				文档 Doc3				得分
	tf(a)	idf	tf-idf	归一化 tf-idf	tf	idf	tf-idf	归一化 tf-idf	
car	1	1.65	1.65	0.560	24	1.65	39.6	0.595	0.92
auto	0.5	2.08	1.04	0.353	0	2.08	0	0	
insurance	1	1.62	1.62	0.550	29	1.62	46.98	0.706	
best	1	1.5	1.5	0.509	17	1.5	25.5	0.383	

于是，在 nnn.atc 下， Score(q,Doc3)>Score(q,Doc1)>Score(q,Doc2)

第七章一个完整搜索系统中的评分计算

习题 **7-3** 给定单个词项组成的查询， 请解释为什么采用全局胜者表（ r=K）已经能够充分保证找到前 K 篇文档。如果只有 s 个词项组成的查询（ s>1），如何对上述思路进行修正？

解答：

词项 t 所对应的 tf 最高的 r 篇文档构成 t 的胜者表。单词项查询， idf 已经不起作用了 (idf 用于区别不同词的先天权重)，所以此时已经足够了。

对于 s 个词项组成的查询，有 idf 权重了。。因此，不再独立。【这一问本人也不知道该怎么答，不扣分吧】

习题 **7-5** 重新考察习题 6-23 中基于 nnn.atc 权重计算的数据，假定 Doc1 和 Doc2 的静态得分分别是 1 和 2。请确定在公式（ 7-2）下，如何对 Doc3 的静态得分进行取值，才能分别保证它能够成为查询 best car insurance 的排名第一、第二或第三的结果。

解答：这道题不扣分吧。 。整个书上有关余弦相似度的计算这块都有问题【即按照公式 (7-2) (6-12)算出的应该是 0 到 1 之间的数，但实际例子 (例 6-4)却是大于 1 的数，例子中都没有考虑查询向量的大小。另外，按照习题 6-23 中 nnn.atc 算出的根本不是什么余弦相似度。整个一团乱】

如果相似度先采用 nnn.atc 计算，最后除以文档向量的大小，则三篇文档的得分分别为： 1.39、 1.47 和 1.68。

- 排名第一： g(d3)+1.68>3.47, g(d3)>1.79
- 排名第二： 2.39< g(d3)+1.68 <3.47, 0.71< g(d3)<1.79
- 排名第三： 0< g(d3) < 0.71

习题 **7-7** 设定图 6-10 中 Doc1、 Doc2 和 Doc3 的静态得分分别是 0.25、 0.5 和 1，画出当使用静态得分与欧几里得归一化 tf 值求和结果进行排序的倒排记录表。

解答：按照公式 7-2 计算得下表：

	doc1	doc2	doc3
car	1.13	0.59	1.58
auto	0.35	1.21	1 (0)
insurance	0.25 (0)	1.21	1.7
best	0.71	0.5 (0)	1.41

所以，倒排记录表如下：

car doc3 doc1 doc2

auto doc2 doc 3 —doc1【按道理， tf 为零的不应该出现在倒排记录中，有的也算对】
insurance doc3 doc2—doc1
best doc3 doc1—doc2

第八章信息检索的评价

习题 **8-8** [*] 考虑一个有 4 篇相关文档的信息需求，考察两个系统的前 10 个检索结果（左边的结果排名靠前），相关性判定的情况如下所示：

系统 1 R N R N N N N N R R

系统 2 N R N N R R R N N N

a. 计算两个系统的 MAP 值并比较大小。

b. 上述结果直观上看有意义吗？能否从中得出启发如何才能获得高的 MAP 得分？

c. 计算两个系统的 R 正确性 值，并与 a 中按照 MAP 进行排序的结果进行对比。

解答：

a. 系统 1 (1+2/3+3/9+4/10)/4=0.6

系统 2 (1/2+2/5+3/6+4/7)/4=0.492

b. 相关文档出现得越靠前越好，最好前面 3-5 篇之内

c. 系统 1 的 R-Precision= 0.5, 系统 2 R-Precision= 0.25

习题 **8-9** [**] 在 10000 篇文档构成的文档集中，某个查询的相关文档总数为 8，下面给出了某系统针对该查询的前 20 个有序结果的相关（用 R 表示）和不相关（用 N 表示）情况，其中有 6 篇相关文档：

R R N N N N N N R N R N N N N R N N N N R

a. 前 20 篇文档的正确率是多少？

P@20=6/20=30%

b. 前 20 篇文档的 F_1 值是多少 ？

R@20=6/8=75%, F1=3/7=0.429

c. 在 25%召回率水平上的插值正确率是多少？

1

d. 在 33%召回率水平上的插值正确率是多少？

3/9=33.3%

e. 假定该系统所有返回的结果数目就是 20，请计算其 MAP 值。

(1+1+3/9+4/11+5/15+6/20)/8=0.4163

假定该系统返回了所有的 10000 篇文档，上述 20 篇文档只是结果中最靠前的 20 篇文档，那么

f. 该系统可能的最大 MAP 是多少？

从第 21 位开始，接连两篇相关文档，此时可以获得最大的 MAP，此时有：

(1+1+3/9+4/11+5/15+6/20+7/21+8/22)/8=0.503

g. 该系统可能的最小 MAP 是多少？

(1+1+3/9+4/11+5/15+6/20+7/9999+8/10000)/8=0.4165

h. 在一系列实验中，只有最靠前的 20 篇文档通过人工来判定，(e)的结果用于近似从 (f)到(g)的 MAP 取值范围。 对于上例来说， 通过 (e)而不是 (f)和 (g)来计算 MAP 所造成的误差有多大 （采用绝对值来计算）？

|0.4163-(0.503+0.4165)/2|=0.043

第九章相关反馈及查询扩展

习题 9-3：用户查看了两篇文档 d1 和 d2，并对这两篇文档进行了判断：包含内容 CDs cheap software cheap CDs的文档 d1 为相关文档，而内容为 cheap thrills DVDs 的文档 d2 为不相关文档。假设直接使用词项的频率作为权重（不进行归一化也不加上文档频率因子），也不对向量进行长度归一化。采用公式（ 9-3）进行 Rocchio 相关反馈，请问修改后的查询向量是多少？其中 $\alpha = 1$ $\beta = 0.75$ $\gamma = 0.25$。

解答：

	cheap	DVDs	extremely	CDs	software	thrills
Query	3	1	1	2	0	0
Doc1	2	0	0	2	1	0
Doc2	1	1	0	0	0	1
Modified Query	4.25	0.75	1	3.5	0.75	0

习题 9-4： Omar 实现了一个带相关反馈的 Web 搜索系统，并且为了提高效率，系统只基于返回网页的标题文本进行相关反馈。用户对结果进行判定，假定第一个用户 Jinxing 的查询是

banana slug

返回的前三个网页的标题分别是：

banana slug Ariolimax columbianus

Santa Cruz mountains banana slug

Santa Cruz Campus Mascot

Jinxing 认为前两篇文档相关，而第 3 篇文档不相关。假定 Omar 的搜索引擎只基于词项频率（不包括长度归一化因子和 IDF 因子）进行权重计算， 并且假定使用 Rocchio 算法对原始查询进行修改， 其中 $\alpha = \beta = \gamma$。请给出最终的查询向量（按照字母顺序依次列出每个词项所对应的分量）。

解答：

	q0	d1	d2	d3	q1
Ariolimax		1			½
banana	1	1	1		2
Campus				1	0
Columbiaus		1			½
Cruz			1	1	0
Mascot				1	0
Moutains			1		½
Santa			1	1	0
slug	1	1	1		2

第十章 XML 检索

(无作业)

第十一章概率检索模型

习题 11-1 根据公式（ 11-18）和公式（ 11-19）推导出公式（ 11-20）。

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t} \qquad (11\text{-}18)$$

	文档	相关	不相关	总计
词项出现	$x_t=1$	s	df_t-s	df_t
词项不出现	$x_t=0$	$S-s$	$(N-df_t)-(S-s)$	$N-df_t$
总计		S	$N-S$	N

(11-19)

$$c_t = K(N, df_t, S, s) = \log \frac{s/(S-s)}{(df_t-s)/((N-df_t)-(S-s))} \qquad (11\text{-}20)$$

解答：代入求解即可。

习题 11-3 令 Xt 表示词项 t 在文档中出现与否的随机变量。假定文档集中有 |R| 篇相关文档，其中有 s 篇文档包含词项 t，即在这 s 篇文档中 Xt=1。假定所观察到的数据就是这些 Xt 在文档中的分布情况。请证明采用 MLE 估计方法对参数 $p_t = (X_t = 1 | R = 1, \phi)$ 进行估计的结果，即使得观察数据概率最大化的参数值为 pt = s/ |R| 。

第十二章基于语言建模的信息检索模型

习题 12-3 习题 12-3 例 12-2 中按照 M1 和 M2 算出的文档的似然比是多少？

解答：由于 P(s|M1) = 0.000 000 000 000 48

, P(s|M2) = 0.000 000 000 000 000 384 , 所以两者的似然比是 0.00000000000048/ 0.000000000000000384 =1250

query	doc1	doc2	doc3	doc4	collection
click	1/2	1	0	1/4	7/16
shears	1/8	0	0	1/4	2/16

习题 12-6 [*] 考虑从如下训练文本中构造 LM：

the martian has landed on the latin pop sensation ricky martin

请问：

a. 在采用 MLE 估计的一元概率模型中, P(the)和 P(martian) 分别是多少？

b. 在采用 MLE 估计的二元概率模型中, P(sensation|pop) 和 P(pop|the) 的概率是多少？

解答：

a.

文档 ID	文档文本
1	click go the shears boys click click click
2	click click
3	metal here
4	metal shears click here

P(the)=2/11, P(martian)=1/11

b. P(sensation|pop)=1, P(pop|the)=0

习题 12-7 [**] 假定某文档集由如下 4 篇文档组成：

为该文档集建立一个查询似然模型。假定采用文档语言模型和文档集语言模型的混合模型，权重均为 0.5。采用 MLE 来估计两个一元模型。计算在查询 click、shears 以及 click shears 下每篇文档模型对应的概率，并利用这些概率来对返回的文档排序。将这些概率填在下表中。

解答：

文档及文档集 MLE 估计

于是，加权以后的估计结果 doc4> doc1>doc2>doc3

第十三章文本分类及朴素贝叶斯方法

习题 **13-2 [*]** 表 13-5 中的文档中, 对于如下的两种模型表示, 哪些文档具有相同的模型表示？哪些文档具有不同的模型表示？对于不同的表示进行描述。 (i) 贝努利模型, (ii) 多项式模型。

第十四章基于向量空间模型的文本分类

第十五章支持向量机及文档机器学习方法

第十六章扁平聚类

第十七章层次聚类

第十八章矩阵分解及隐性语义索引

第十九章　**Web** 搜索基础

第二十章　**Web** 采集及索引

第二十一章链接分析