第一次作业（更新版本，3.18 周一上课交）

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？

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。

习题 2-4 [*] 对于（2-1）中所示 Porter 工具中使用的规则集：

a. 为什么要引入一条自己到自己的规则：SS→SS?

b. 采用该规则集进行处理，请给出如下词的词干还原结果。

circus canaries boss

c. 要对 pony 进行正确的词干还原处理，应该增加一条什么规则？

d. 对 ponies 和 pony 进行词干还原处理的结果看上去可能很怪异，这会对检索的结果造成负面影响吗？为什么？

规则		示例		
SSES	→SS	caresses	→caress	
IES	→I	ponies	→poni	(2-1)
SS	→SS	caress	→caress	
S	→	cats	→cat	

习题 2-6 [*] 对于两个词组成的查询，其中一个词（项）的倒排记录表包含下面 16 个文档 ID：

[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180]

而另一个词（项）对应的倒排记录表仅仅包含一个文档 ID：

[47]

请分别采用如下两种策略进行倒排记录表合并并计算所需要的比较次数，同时简要地说明计算的正确性。

a. 使用标准的倒排记录表。

b. 使用倒排记录表+跳表的方式，跳表指针设在 $\sqrt{P}$ 处。

习题 2-10 [*] 考虑如下位置索引的片段，格式为：词: 文档: 〈位置, 位置, . . .〉; 文档: 〈位置, . . .〉。

. . .

Gates: 1: 〈3〉; 2: 〈6〉; 3: 〈2,17〉; 4: 〈1〉;

IBM: 4: 〈3〉; 7: 〈14〉;

Microsoft: 1: 〈1〉; 2: 〈1,21〉; 3: 〈3〉; 5: 〈16,22,51〉;

/k 操作符的意思是，当查询为词 1 /k 词 2 时表示查找时词 1 和词 2 必须在 k 个词内出现（匹配时左右两边计算都行），其中参数 k 是一个正整数。若 k = 1，则意味着词 1 和词 2 相邻。

a. 给出满足查询 Gates /2 Microsoft 的所有文档。

b. 将 k 值划分成多个集合，使得同一集合内的 k 值对于查询 Gates /k Microsoft 返回同样的文档子集，而不同集合中的 k 值则返回不同的文档子集。

习题 4-2 [*] 在基于块的排序索引算法中，如何快速构建词典来避免对数据的额外扫描？

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？

表4-1　一个2007年度的典型计算机系统的参数

符　号	含　义	值
s	平均寻道时间	$5\text{ms} = 5\times10^{-3}\text{s}$
b	每个字节的传输时间	$0.02\mu\text{s} = 2\times10^{-8}\text{s}$
	处理器时钟频率	10^9Hz
p	底层操作时间（如字的比较或者交换）	$0.01\mu\text{s} = 10^{-8}\text{s}$

（续）

符　号	含　义	值
	内存大小	几吉字节
	磁盘大小	1 TB或更大

注：寻道时间指的是将磁头移到新位置所花的时间。每个字节所需的传输时间指的是磁头就位以后从磁盘到内存的传输速率。

习题 4-4　给定 $n=2$ 及 $1\leqslant T\leqslant 30$，对图 4-7 的算法进行逐步模拟。画出一个表格，给出在给定 $T=2^*k$ 个词条已处理点时（$1\leqslant k\leqslant 15$）所用到的 $I_0,\cdots,I_3$ 中的索引。该表格的前三行如下：

	I_3	I_2	I_1	I_0
2	0	0	0	0
4	0	0	0	1
6	0	0	1	0

```
LMERGEADDTOKEN(indexes, Z0, token)
 1  Z0 ← MERGE(Z0, {token})
 2  if |Z0| = n
 3    then for i ← 0 to ∞
 4         do if Ii ∈ indexes
 5              then Zi+1 ← MERGE(Ii, Zi)
 6                   (Zi+1 is a temporary index on disk.)
 7                   indexes ← indexes − {Ii}
 8              else Ii ← Zi   (Zi becomes the permanent index Ii.)
 9                   indexes ← indexes ∪ {Ii}
10                   BREAK
11         Z0 ← ∅

LOGARITHMICMERGE()
1  Z0 ← ∅   (Z0 is the in-memory index.)
2  indexes ← ∅
3  while true
4  do LMERGEADDTOKEN(indexes, Z0, GETNEXTTOKEN())
```

图 4-7　对数合并示意图，其中每个词条（词项 ID, 文档 ID）一开始通过 LMERGEADDTOKEN 函数加入到内存索引 Z_0 中。LOGARITHMICMERGE 对 Z_0 和索引初始化

习题 4-10　为了对负载均衡进行优化，MapReduce 中的倒排器必须获得大小相近的分区倒排记录文件，但是对于一个新文档集，事先关不知道键-值对的分布，应该如何解决这个问题？