HW1

裴启智

PB18111793

计算题

1.1

1.1 请推荐如下查询的处理次序。
(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	213 312
kaleidoscope	87 009
marmalade	107 913
skies	271 658
tangerine	46 653
trees	316 812

先估计每个OR操作后的结果大小

(tangerine OR trees) 约为 46653 + 316812 = 363465

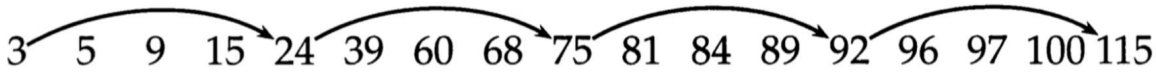
(marmalade OR skies)约为 107913 + 271658 = 379571

(kaleidoscope OR eyes)约为 87009 + 213312 = 300321

故应该按照从小到大的顺序，先处理 (tangerine OR trees) AND (kaleidoscope OR eyes)，得到结果后（假设为A）再处理A AND (marmalade OR skies)

1.2

1.2 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
采用基于跳表指针的倒排记录表合并算法，请问：
1) 跳表指针实际发生跳转的次数是多少？
2) 当两个表进行合并时，倒排记录之间的比较次数是多少？
3) 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？

带有跳表指针时，算法执行过程如下：（假设表A含跳表指针，表B不含跳表指针）

```
INTERSECTWITHSKIPS(p1,p2)
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 if hasSkip(p1) and (docID(skip(p1))≤docID(p2))
9                 then while hasSkip(p1) and (docID(skip(p1))≤docID(p2))
10                     do p1← skip(p1)
11                else p1←next(p1)
12           else if hasSkip(p2) and (docID(skip(p2))≤docID(p1))
13                then while hasSkip(p2) and (docID(skip(p2))≤docID(p1))
14                     do p2←skip(p2)
15                else p2←next(p2)
16   return answer
```


1. 比较3 = 3，发现共同的记录3
2. A、B表指针同时向后移动一位
3. 比较5 = 5，发现共同的记录5
4. A、B表指针同时向后移动一位
5. 比较9 < 89, A表指针向后移动一位
6. 比较15 < 89,A表指针向后移动一位
7. 比较24 < 89,比较75 < 89, A表指针跳到75
8. 比较92 > 89,故不跳表
9. 比较75<89,比较92>89,故不跳表，A表指针向后移动一位
10. 比较81 < 89,A表指针向后移动一位
11. 比较84 < 89,A表指针向后移动一位
12. 比较89 = 89,发现共同的记录89
13. A、B表指针同时向后移动一位
14. 比较92 < 95,比较115 > 95, 故不跳表,A表指针向后移动一位
15. 比较 96 > 95, B表指针向后移动一位
16. 比较96 < 97,A表指针向后移动一位
17. 比较97 = 97, 发现共同记录97
18. A、B表指针同时向后移动一位
19. 比较100 > 99, B表指针向后移动一位
20. 比较100 = 100,发现共同记录100
21. A、B表指针同时向后移动一位
22. 比较115 > 101,B表指针向后移动一位
23. 循环结束

1)实际发生跳转的次数为1（24->75)

2)比较次数20

3)不使用跳表指针,则流程如下

1. 比较3 = 3，发现共同的记录3

2. A、B表指针同时向后移动一位
3. 比较5 = 5，发现共同的记录5
4. A、B表指针同时向后移动一位
5. 比较9 < 89, A表指针向后移动一位
6. 比较15 < 89,A表指针向后移动一位
7. 比较24 < 89,A表指针向后移动一位
8. 比较39 < 89, A表指针向后移动一位
9. 比较60 < 89,A表指针向后移动一位
10. 比较68 < 89,A表指针向后移动一位
11. 比较75 < 89,A表指针向后移动一位
12. 比较81 < 89,A表指针向后移动一位
13. 比较84 < 89,A表指针向后移动一位
14. 比较89 = 89,发现共同的记录89
15. A、B表指针同时向后移动一位
16. 比较92 < 95,A表指针向后移动一位
17. 比较 96 > 95, B表指针向后移动一位
18. 比较96 < 97,A表指针向后移动一位
19. 比较97 = 97, 发现共同记录97
20. A、B表指针同时向后移动一位
21. 比较100 > 99, B表指针向后移动一位
22. 比较100 = 100,发现共同记录100
23. A、B表指针同时向后移动一位
24. 比较115 > 101

比较次数为19

1.3

1.3 写出倒排记录表（777, 17743, 294068, 31251336）的可变字节编码。在可能的情况下对间距而不是文档 ID 编码。写出 8 位块的二元码。

文档ID	**777**	**17743**	**294068**	**31251336**
间距		16966	276325	30957268
VB编码	00000110 10001001	00000001 00000100 11000110	00010000 01101110 11100101	00001110 01100001 00111101 11010100

问答题

2.1

基于机械分词的常见方法中对于“最大匹配”的依赖，可能导致什么隐患？如何利用 N-最短路径缓解这一隐患？如何选择一个恰当的 N 值

隐患：对于有歧义或者多种切分方式的句子，利用“最大匹配”可能无法得到正确的结果，同时，需要耗费较大的开支来维护高质量的词典，且无法应对新生词汇，词频和词汇的重要性对结果一般不会产生影响。

利用N-最短路径，保留N条最短的路径，提供更多的分词方案，便于消除歧义。另外可以在N-最短路径算法中加入边权重，这样就可以实现基于统计的分词方法，结合词频和词汇重要性进行分词。

N值的选择：

1. 可以通过构造关于分词的训练集，对大量的句子的分词方法进行人工标注，然后通过机器学习、深度学习技术，对包含N-最短路径算法的模型进行训练，确定N-最短路径算法中的参数N
2. 也可先设置较大的N，进行试探，再从试探的结果中人工寻找正确的切分方法，以此来标注数据，从而不断缩小设置的N值
3. 基于统计的方法，根据大量已有的分词及其选取的N条最短路径，计算平均来确定N值

2.2

如何结合查询词项的分布细节，设计相对合理的跳表指针步长？

考虑A AND B

如果A对应表中Web文档的ID号分布密度较小，即两个文档ID间的差值较大，而B对应表中Web文档的ID号分布密度较大，则可在B中设置相对较大的跳表指针

如果A对应表中Web文档的ID号集中分布在a附近，在b附近也少量分布，而B对应表中Web文档的ID号集中分布b附近，在a附近几乎没有分布，则可设置A的跳表指针直接跳过a附近的区域

如果A、B对应的倒排表中文档的ID都在同一区间段比较密集，则应设置步长相对较小的跳表指针

如果A、B对应的倒排表中文档的ID都在同一区间段比较稀疏，则应少设或者不设跳表指针

2.3

在信息检索系统中，如何**同时使用**位置索引和停用词表？潜在问题有哪些，如何解决？

一般而言停用词不是我们需要检索的核心词汇，所以可直接根据停用词表边过滤，边构建位置索引。这样可以过滤掉大量没有实际意义的停用词，降低倒排表的存储空间。（停用词的位置索引一般很多）

当然，对于已经建立好的未过滤停用词的位置索引，也可通过停用词表，删除那些停用词对应的位置索引

潜在问题：

- 空间：

  存储位置索引对存储空间的要求更大，位置索引在存储文档ID的同时还要存储其位置信息
- 停用词意义上：

  并不能完全删除停用词，因为某些停用词的组合有意义，或者这些停用词在特定场景下有意义

解决：

- 空间

  采用索引压缩技术，如可以采用可变长度编码，存储位置间的距离，而不是存储具体的位置。

- 停用词意义上：
  - 建立由有意义的停用词词组构成的词典，过滤时要同时结合该词典，以免过滤掉一些具有实际意义的停用词
  - 或者将这些有意义的停用词词组视作二元或多元词加入到词汇表中，这样就可以在建立倒排表时建立关于这些词组的位置索引，而去掉那些无意义的停用词