Due: 2019/3/8

Homework 1

Student Number: 118033910019
Name: Xingyi Wang

Problem 1. (20 points) The following pairs of words are stemmed to the same form by the Porter stemmer. Which pairs would you argue shouldn't be conflated. Give your reasoning.

a. abandon/abandonment

b. absorbency/absorbent

c. marketing/markets

d. university/universe

e. volume/volumes

Solution.

a. abandon and abandonment can be stemmed into abandon.

b. absorbency and absorbent can be stemmed into absorb.

c. marketing and markets can be stemmed into market.

d. university and universe should not be conflated because these two words have different meanings.

e. volume and volumes can be stemmed into volume.

Problem 2. (30 points)

Doc 1: new home sales top forecasts

Doc 2: home sales rise in july

Doc 3: increase in home sales in july

Doc 4: july new home sales rise

Consider the documents above,

a. Draw the term-document incidence matrix for this document collection.

b. Draw the inverted index representation for this collection.

c. For the document collection, what are the returned results for these queries:

  i july AND rise

  ii (NOT increase) AND (home OR sale)

Solution.

a. The term-document incidence matrix is shown below:

	Doc 1	Doc 2	Doc 3	Doc 4
forecast	1	0	0	0
home	1	1	1	1
in	0	1	1	0
increase	0	0	1	0
july	0	1	0	1
new	1	0	0	1
rise	0	1	1	1
sale	1	1	1	1
top	1	0	0	0

b. The inverted index representation is shown below:
forecast: 1
home: 1 → 2 → 3 → 4
in: 2 → 3
increase: 3
july: 2 → 4
new: 1 → 4
rise: 2 → 3 → 4
sale: 1 → 2 → 3 → 4
top: 1

c. i The returned result for “july AND rise” is Doc 2 and Doc 4.

  ii The returned result for “(NOT increase) AND (home OR sale)” is Doc 1, Doc 2, and Doc 4.

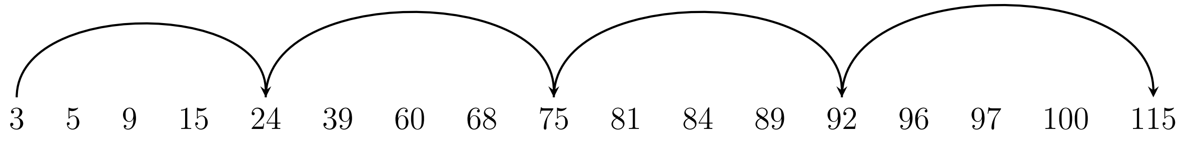
Problem 3. (30 points) Write out a postings merge algorithm, in the style of Algorithm 1, for an x OR y query.

Solution.

Algorithm 1: MERGE(p_1, p_2)

```
$answer \leftarrow ()$
while $p_1 \neq NIL$ and $p_2 \neq NIL$ do
    if $docID(p_1) = docID(p_2)$ then
        ADD($answer$, $docID(p_1)$)
        $p_1 \leftarrow next(p_1)$ $p_2 \leftarrow next(p_2)$
    else
        if $docID(p_1) < docID(p_2)$ then
            ADD($answer$, $docID(p_1)$)
            $p_1 \leftarrow next(p_1)$
        else
            ADD($answer$, $docID(p_2)$)
            $p_2 \leftarrow next(p_2)$
while $docID(p_1) \neq NIL$ do
    ADD($answer$, $docID(p_1)$)
    $p_1 \leftarrow next(p_1)$
while $docID(p_2) \neq NIL$ do
    ADD($answer$, $docID(p_2)$)
    $p_2 \leftarrow next(p_2)$
return $answer$
```

Problem 4. (30 points) Consider a postings intersection between this postings list, with skip pointers:



and the following intermediate result postings list (which hence has no skip pointers):
3 5 89 95 97 99 100 101
Trace through the postings intersection algorithm(pdf of lecture 1, page 39)

a. How often is a skip pointer followed?

b. How many postings comparisons will be made by this algorithm while intersecting the two lists?

c. How many postings comparisons would be made if the postings lists are intersected without the use of skip pointers?

Solution.

a. In the postings list, there is 1 skip pointer out of every 4 postings. And only 1 skip pointer is used during the postings intersection algorithm.

b. The comparisons are listed below:
3-3, 5-5, 9-89, 15-89, 24-89, 75-89, 92-89, 81-89, 84-89, 89-89, 92-95, 96-95, 96-97, 97-97, 100-99, 100-100, 115-101. So 17 comparisons would be made.

c. If the postings lists are intersected without the use of skip pointers, the comparisons are listed below:
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. So 19 comparisons would be made.