https://miro.com/app/board/o9J_lPlt-q8=/

The Term Vocabulary + Postings list (Chapter 2)

Exercise 2/1

Are the following statements true or false?

F 1. In a Boolean retrieval system, **stemming never lowers precision**.
T 2. In a Boolean retrieval system, **stemming never lowers recall**.
F 3. Stemming **increases the size** of the vocabulary.
F 4. Stemming **should** be invoked **at indexing time** but not while processing a query.

Definition 1 (Recall)
Recall describes how many of the relevant documents are retrieved.

$$recall = R = \frac{\#relevant\ retrieved}{\#relevant}$$

Definition 2 (Precision)
Precision describes how many of the retrieved documents are relevant.

$$precision = P = \frac{\#relevant\ retrieved}{\#retrieved}$$

user wants (relevant docs) — you give (retrieved docs)

Definition 3 (Porter stemmer)
The entire Porter's algorithm is too complex to present here. It consists of 5 phases of word reductions, applied sequentially. The first phase uses the following rule group. Importantly, only the rule that applies to the longest suffix is used.

Rule			Example		
SSES	→	*SS*	*caresses*	→	*caress*
IES	→	*I*	*ponies*	→	*poni*
SS	→	*SS*	*caress*	→	*caress*
S	→		*cats*	→	*cat*

$Y \rightarrow I$

pony --> poni

ponies --> poni

Exercise 2/4

For the **Porter stemmer group** shown in Definition 3:

1. What is the **purpose** of including an identity rule such as SS → SS?
2. Applying just this rule group, what will the following words be stemmed to?: **circus, canaries, boss**
3. What rule should be added to correctly stem **pony**?
4. The stemming for pony and ponies might seem strange. Does it have a deleterious **effect on retrieval**? Why or why not?

circus -> circu

canaries -> canari

boss -> boss

Exercise 2/5

Below is a part of **index with positions** in the form **doc1**: (pos1, pos2, pos3, . . .); **doc2**: (pos1, pos2, . . .); . . .

1. **angels**: **2** : (36, 174, 252, 651); **4** : (12, 22, 102, 432); **7** : (17);
2. **fools**: **2** : (1, 17, 74, 222); **4** : (8, 78, 108, 458); **7** : (3, 13, 23, 193);
3. **fear**: **2** : (87, 704, 722, 901); **4** : (13, 43, 113, 433); **7** : (18, 328, 528);
4. **in**: **2** : (3, 37, 76, 444, 851); **4** : (10, 20, 110, 470, 500); **7** : (5, 15, 25, 195);
5. **rush**: **2** : (2, 66, 194, 321, 702); **4** : (9, 69, 149, 429, 569); **7** : (4, 14, 404);
6. **to**: **2** : (47, 86, 234, 999); **4** : (14, 24, 774, 944); **7** : (19, 319, 599, 709);
7. **tread**: **2** : (57, 94, 333); **4** : (15, 35, 155); **7** : (20, 320);
8. **where**: **2** : (67, 124, 393, 1001); **4** : (11, 41, 101, 421, 431); **7** : (15, 35, 735);

The following terms are phrase queries. Which documents correspond to the following queries and on which positions?

The index is incorrect. How? (hint: what properties must each index have?)

Query1: fools rush in

doc2: <1, 2, 3>, <17,18,19>, <74,75,76>, ... — this is what we want

doc2: <1, 2, 3>, <17,66,37>, <74,194,76>, ... — this is what we get

doc2: <1, 2, 3>; doc4: <8, 9, 10>; doc7: <3, 4, 5>, <13, 14, 15> — final result

Query2: fools rush in AND angels fear to tread

Exercise 2/9

1. List the **comparisons** performed **to intersect** the following sorted non-positional postings lists with skip pointers of frequency 5.

P1 = [2, 10, 12, 16]

(2,1), (2,7), (2,3), (10,3), (10,9), 10,5),
(10,6), (10,7), (10,12), (10,8), (10,9), (10,10), (12,11), (12,12),
(16,13), (16,14), (16,15)

P2 = [1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]