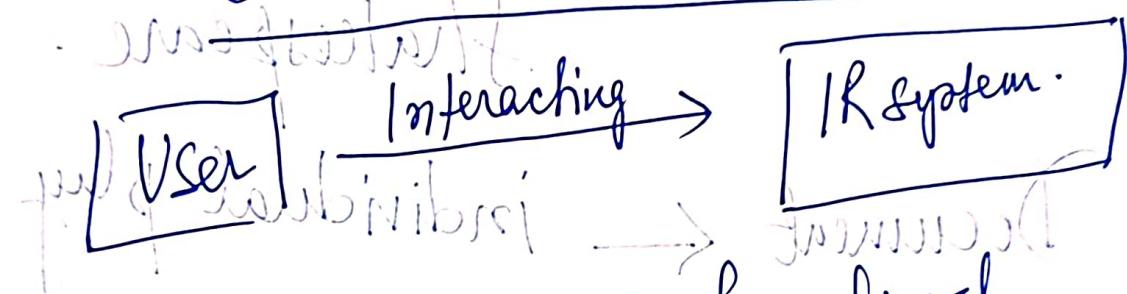


Information Retrieval Boolean Retrieval



User — unit of retrieval.

Document — each document is described using

Doc ID — each document is identified by a Doc ID.

Collection — collection of documents

User — has an information need.

User — query : word, phrase
word, phrase combined with operator feed to IR System

query

Term — unit of information (word / phrase).

IR System — tasked to retrieving relevant documents.

Corpus ← All plays written

Shakespeare

Document ← individual play

Query \leftarrow a Boolean expression of terms connected by \wedge .

Boolean operators :

AND, OR, NOT. etc.

~~Ques~~ Query: Which all plays of Shakespeare contain the words Brutus AND Caesar but NOT Calpurnia.

plays containing ~~Caesar~~ ~~Julius Caesar~~ ~~Caesar~~

② Go line by line & omit those that have Calpurnia.

Why does it not work this way?

①

Very slow

initial

suffix

word

inflected

derived

present

past

-Xidew ② ambiguous ③ burst w/t

~~NOT Calpurnia~~

Roman's

near

within three words

ranked retrieval

top of list

high priority

high weight

high value

is slowing down things

Countrymen.

.

.

.

.

.

.

.

The term-incident matrix.

	Document	The Taming of the Shrew	Hamlet	Othello	Macbeth
Character					
Antonio	1	0	0	0	1
Brutus	1	0	0	0	0
Caesar	1	0	0	0	0
Cleopatra	1	0	0	0	0
Calpurnia	0	0	0	0	0
Horatio	0	1	1	1	0
Portia	0	0	1	1	0
Servant	0	1	1	1	0

BRUTUS AND CAESAR NOT CALPURNIA. }
 [11010] & [110110] ^ v[010000]. }
 response: [000000]. }

- Antonio & Cleopatra → Act III, Scene ii

- Hamlet → Act III, Scene ii

Reductionist

Holistic ("more ^{is} different").

What is the problem?

Bigger collections

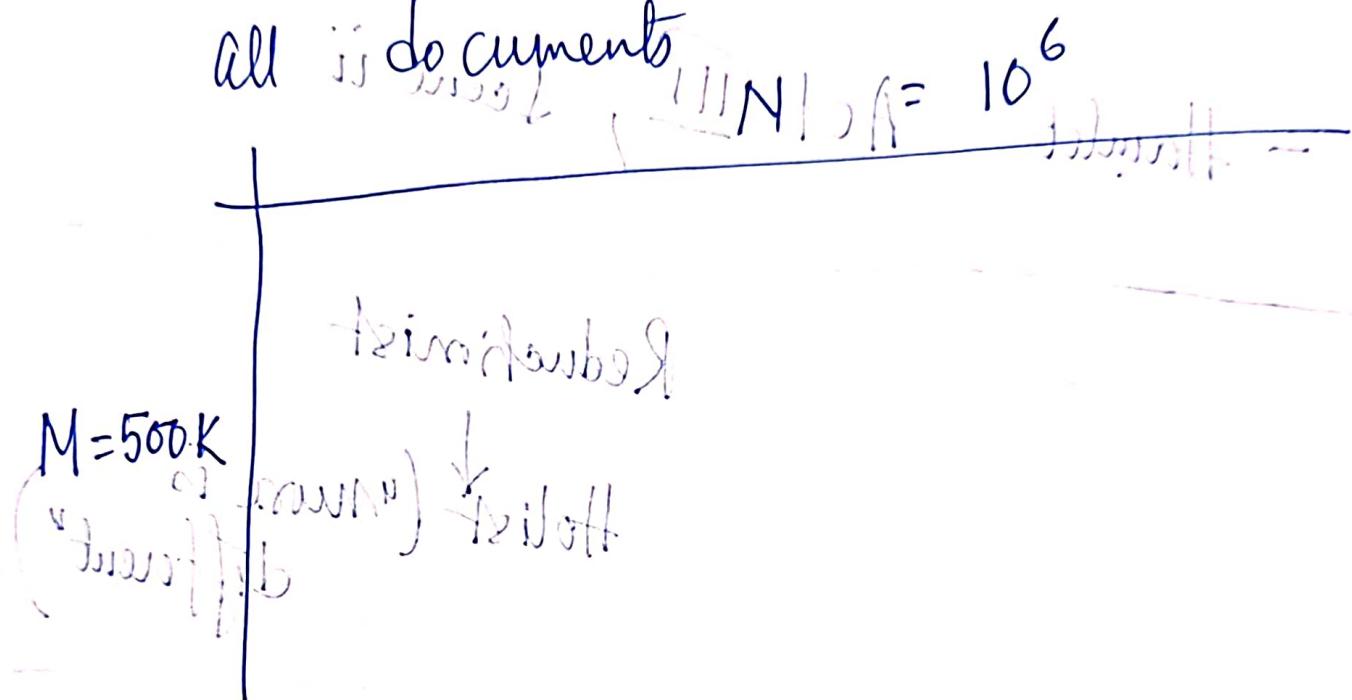
$N = 1 \text{ M}$ documents; 1 G

Each doc \leftarrow avg \rightarrow 1000 words

$\Rightarrow 6 \text{ bytes} / \text{word} \times 1000 \text{ words}$

How ~~much~~ much data on documents? 6 GB.

if we let $M = 500K$ and distinct terms across all its documents, $|UNI| \approx 10^6$



$\frac{1}{2}$ trillion entries will be zero

Engineer's  sparse matrix.

Every column will have only 1000 s.

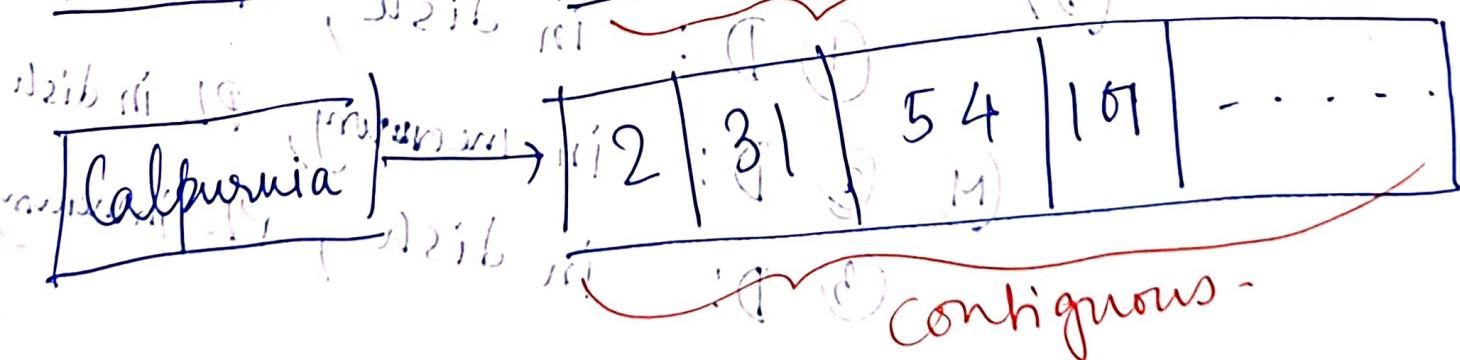
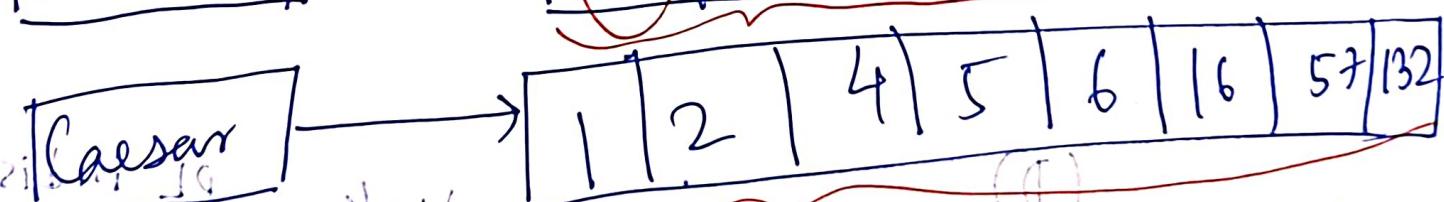
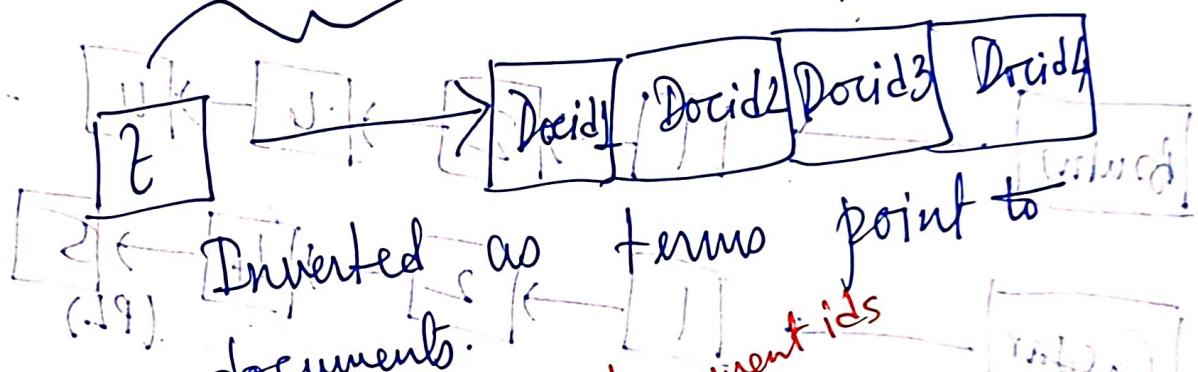
Rest every thing is zero.

How many entries have info? 1 b.

Ans: 1. dimension and document matrix

Inverted Index

For each term t we store a list of all documents that contain t



To keep it

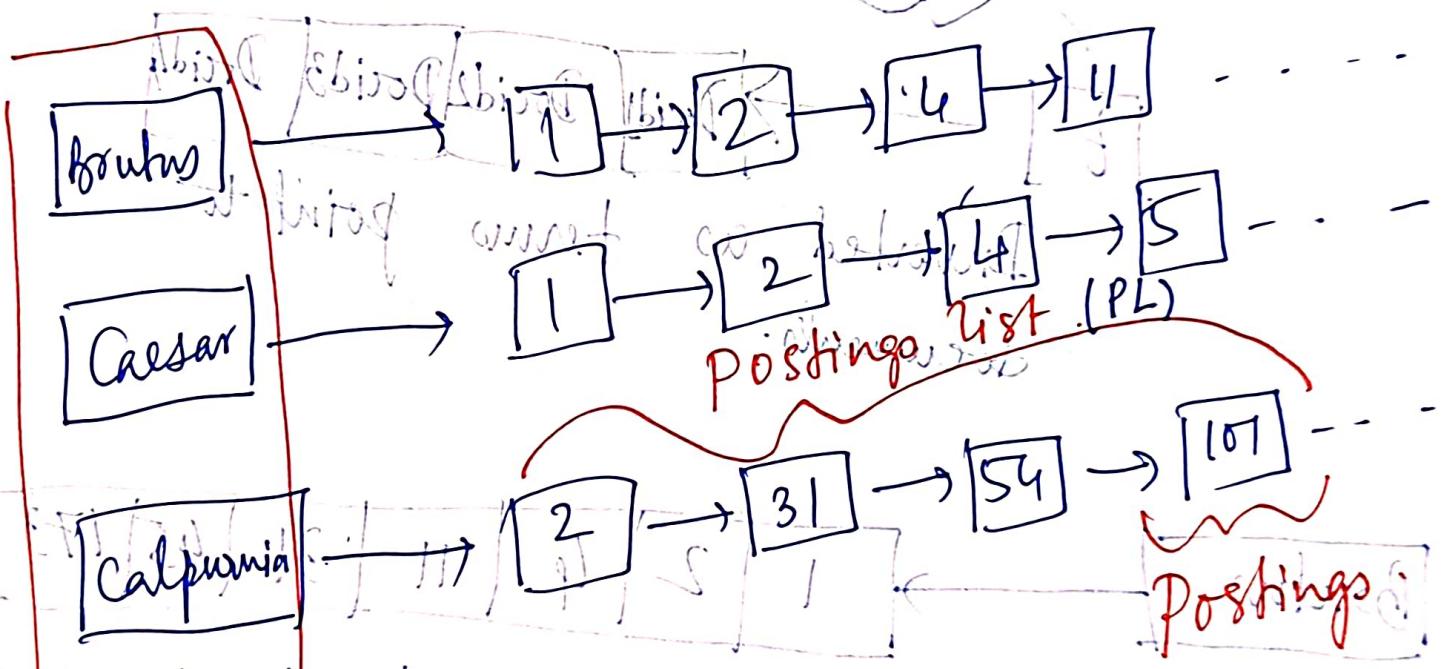
Contiguous

or not contiguous

list (array)

if insertion

arrival of new docs?



Dictionary (D)

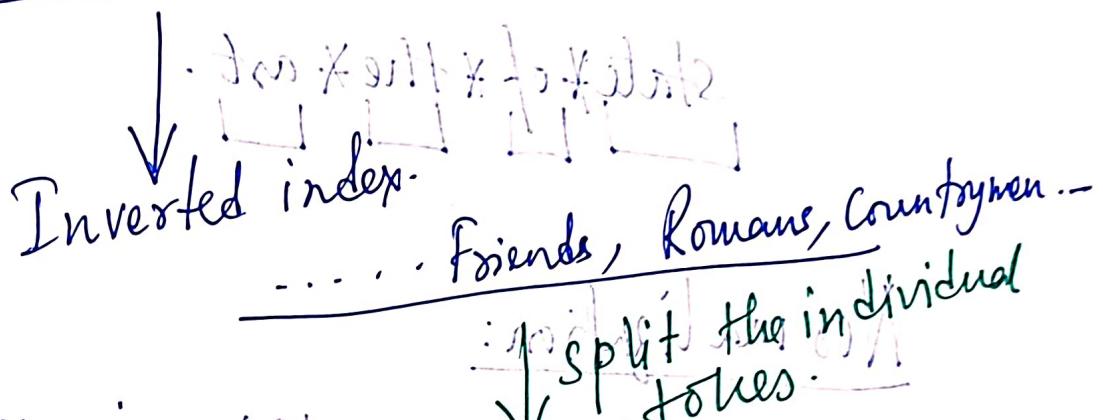
① D: in disk,

(M)

② D: in memory, PL in disk

③ D: in disk, PL in memory

Raw text & documents classification



Tokenizer



mid punctuation, stop words

friend

roman

countrymen

Stems of a word.

stem without stop words
stemming

Indexer

indexing
with respect to
content of document

import part of

Inverted index

Tokenization

state of the art.

Normalization:

U.S.A. ≡ USA.

Stemming:

authorize, authorization

should map to a single term.

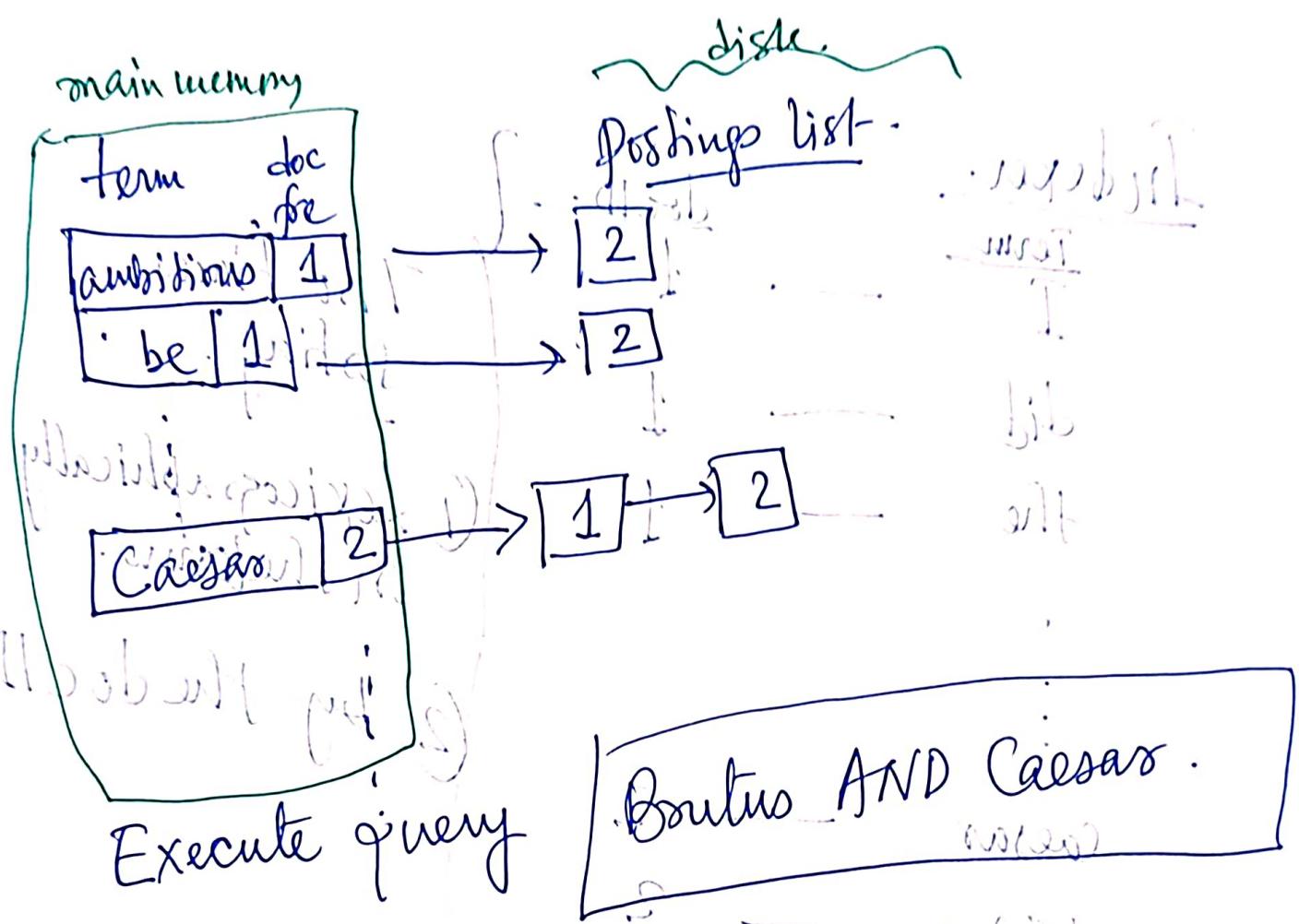
Stop words / function words.

the, a, to, of

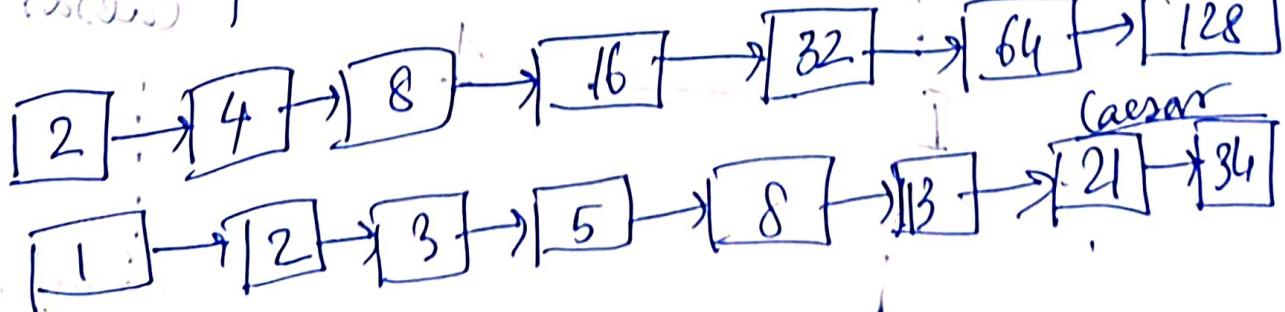
Linguistically no semantics

→ render
syntax
& semantics
to other
words.

→ very frequent.



- S - Step 1: Locate Boutus in dictionary
- E - 2:nd Retrieve postings list of Boutus from disk.
- S - 3:rd Locate Caesar in dictionary
- E - 4:th Retrieve postings list of Caesar from disk.



Merge ~~Sort~~ Algorithm (Recall merge sort)

List 1 - $|x|$

List 2 $\rightarrow |y|$

Complexity:

$O(|x| + |y|)$ Hence we kept the lists sorted.

int INTERSECT (P_1, P_2) →

1. answer $\leftarrow \langle \rangle$

2. while $P_1 \neq \text{NIL}$ & $P_2 \neq \text{NIL}$

3. do if docID(P_1) = docID(P_2),

4. then ADD(answer, docID(P_1))

5. $P_1 \leftarrow \text{next}(P_1)$

6. $P_2 \leftarrow \text{next}(P_2)$

7. else if docID(P_1) < docID(P_2)

8. then $P_1 \leftarrow \text{next}(P_1)$

9. else $P_2 \leftarrow \text{next}(P_2)$

10. return answer.

Any document as a Set of terms.

→ ~~Ranking~~ → ~~ticks~~
← ~~spill~~

The ~~if~~ most popular + retrieval system - last 3 decades.

Mac + OS X spotlight → still uses this

Westlaw: <http://www.westlaw.com/>

legal experts with

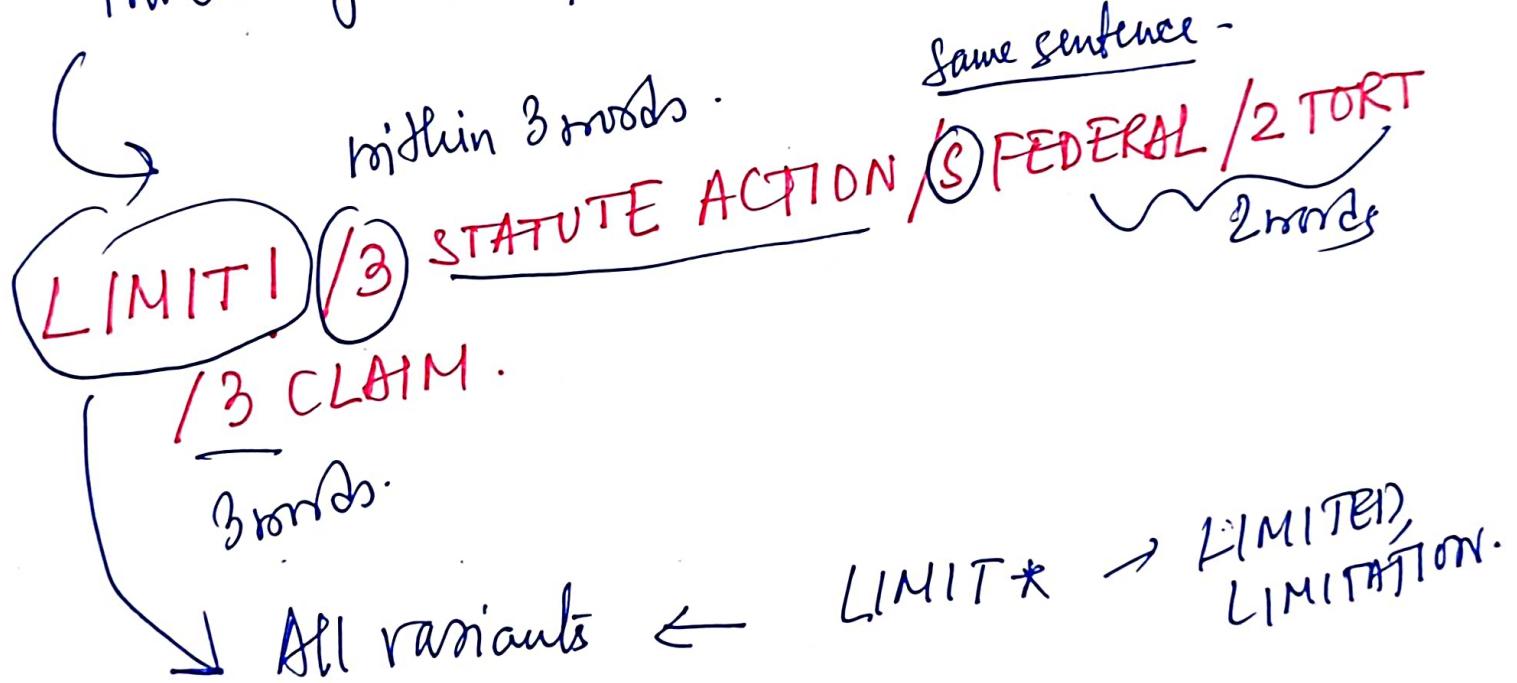
700.000 users.

Started → 1975.

Ranking → 1992

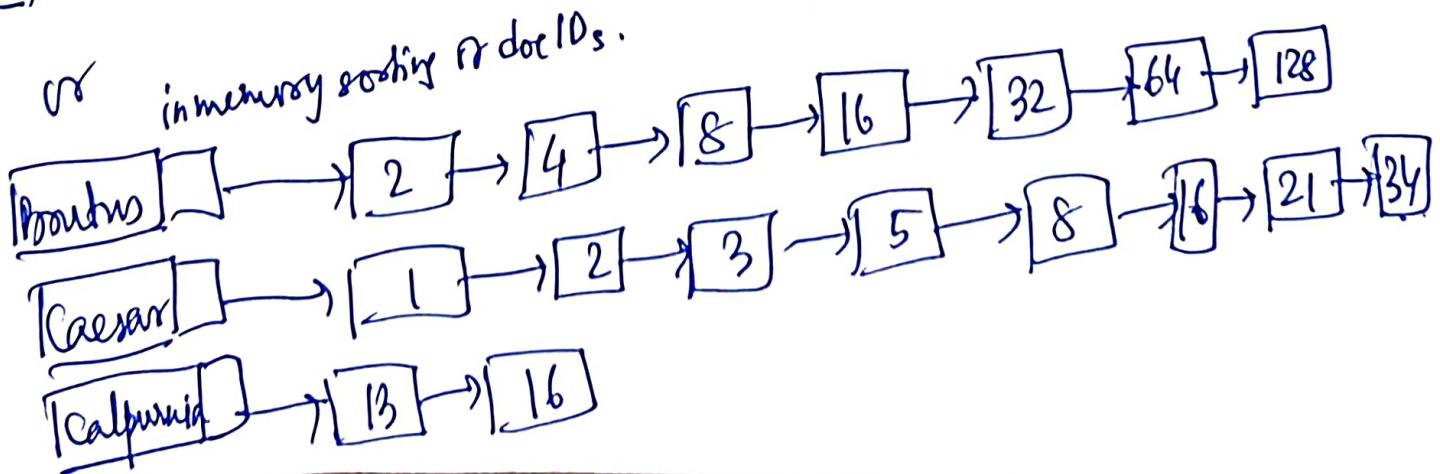
federated search / parallel search) → 2010

Query:
What is the statute of limitations in cases involving the federal tort claims act?

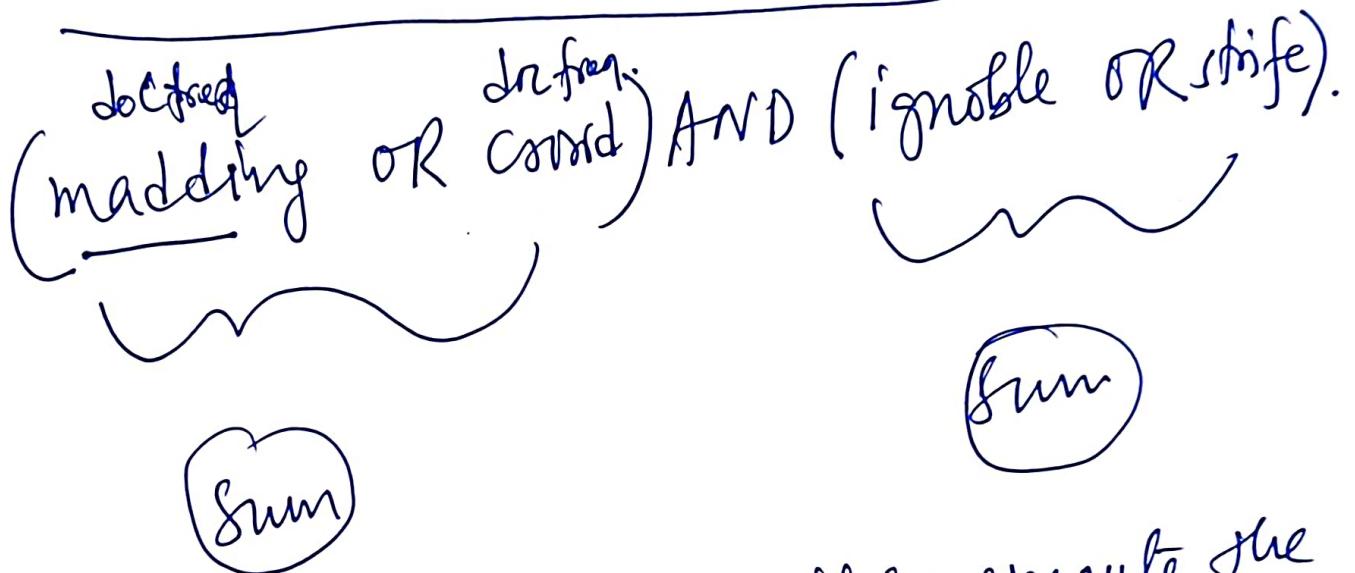


~~Brutus AND CAF~~

Execute : Brutus AND Calpurnia AND Caesar



What about OR queries?



order by the sum & then execute the ANDs.

Google search \rightarrow Boolean queries-

$$[w_1, w_2, \dots, w_n] \longrightarrow w_1 \text{ AND } w_2 \dots \text{ AND } w_n$$

When does it fail?

- very few docs.
- all w_i s are unfortunately in the anchor text
- all ~~unique~~ words present are variants of one