Information Retrieval

Prof: Ehab Ezzat Hassanein

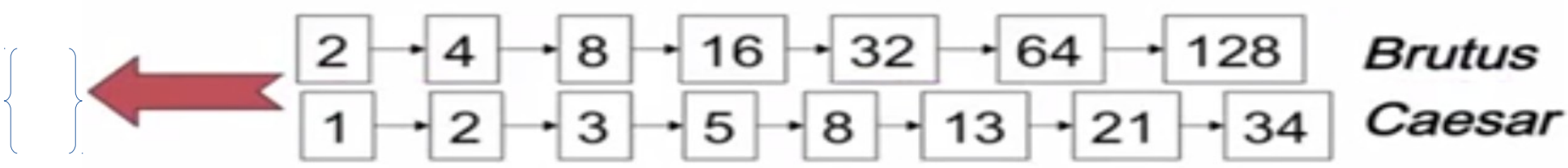
Query Processing Inverted Indexes

Query Processing: AND

- Consider Processing query:

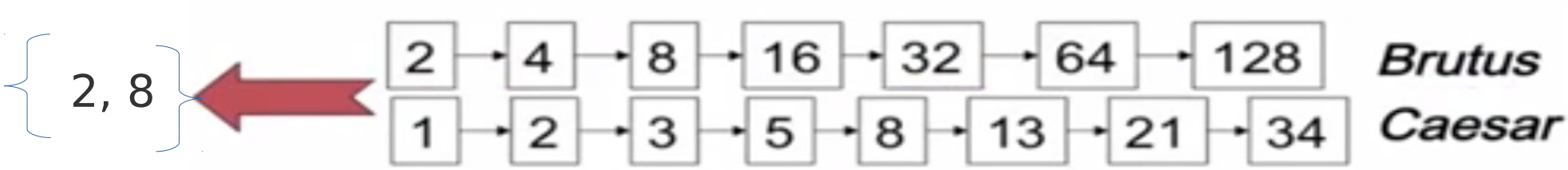
  Brutus and Caesar

- 1. Locate Brutus in the Dictionary
- 2. Retrieve its postings
- 3. Locate Caesar in the Dictionary
- 4. Retrieve its postings
- 5. Merge the two postings lists (intersect the document sets):



Algorithm for the merging of two postings lists

- Walk through the two postings simultaneously, in time linear in the total number of postings entries.
- If the list lengths are x any y, the merge takes O(x + y) operations
- Formally, the complexity of querying is **Θ (N)** , *where N is the number of documents in the collection*
- Crucial: postings sorted by DocId



Algorithm for the merging of two postings lists

```
INTERSECT(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 p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Boolean Retrieval Model & Extended Boolean models

Boolean queries: Exact match

- The Boolean Model Retrieval is being able to ask a query that is a Boolean Expression:
  - Boolean queries using **AND, OR and NOT** to join query terms
    - View each Document as a set of words
    - Is precise: document matches condition or not
  - Perhaps the simplest model to build an IR system on
  - The primary commercial retrieval tool for 3 decades.
    - Still used in: Email, library catalog, Mac OS X Spotlight

Extended Retrieval Model

- The ***Extended Boolean model*** was described in a Communications of the ACM article appearing in 1983, by Gerard Salton, Edward A. Fox, and Harry Wu.
- The goal of the Extended Boolean model is to overcome the drawbacks of the Boolean model that has been used in information retrieval.
- The Boolean model doesn't consider term weights in queries, and the result set of a Boolean query is often either too small or too big.

Example: WestLaw

Largest commercial (paying subscribers) legal search service (started 1975; ranking added 1992; new federal search added 2010)

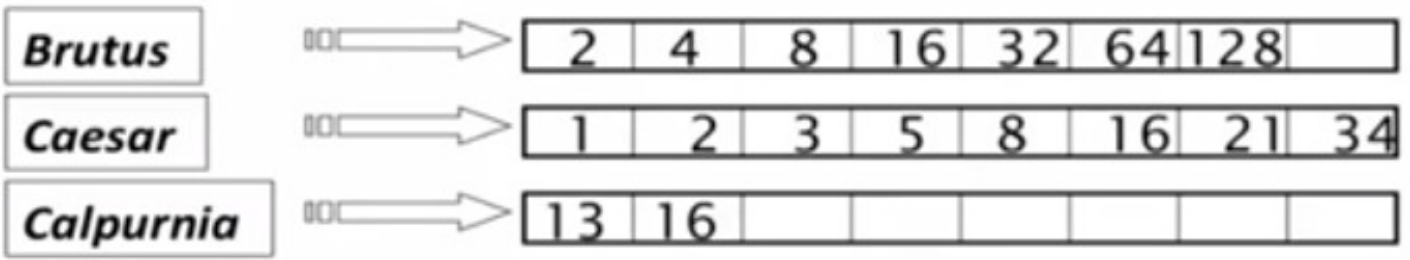
- Tens of terabytes of data; ~700,000 users
- Majority of users still use Boolean queries
- Example query:
  - What is limitations in case involving the federal tort claims act?
  - LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
    - /3 = within 3 words, /S = in the same sentence

Example: WestLaw

- Another example query:
  - Requirements for disabled people to access a work place
  - disabl! /p access! /s work-site work-place (employment /3 place
- Note that SPACE is disjunction not conjunction
- Long, precise queries; proximity operators; incrementally developed; not like web search
- Many Professional searchers still like this extended Boolean search
  - You know exactly what you are getting
- But that doesn’t mean it actually work better

Query Optimization

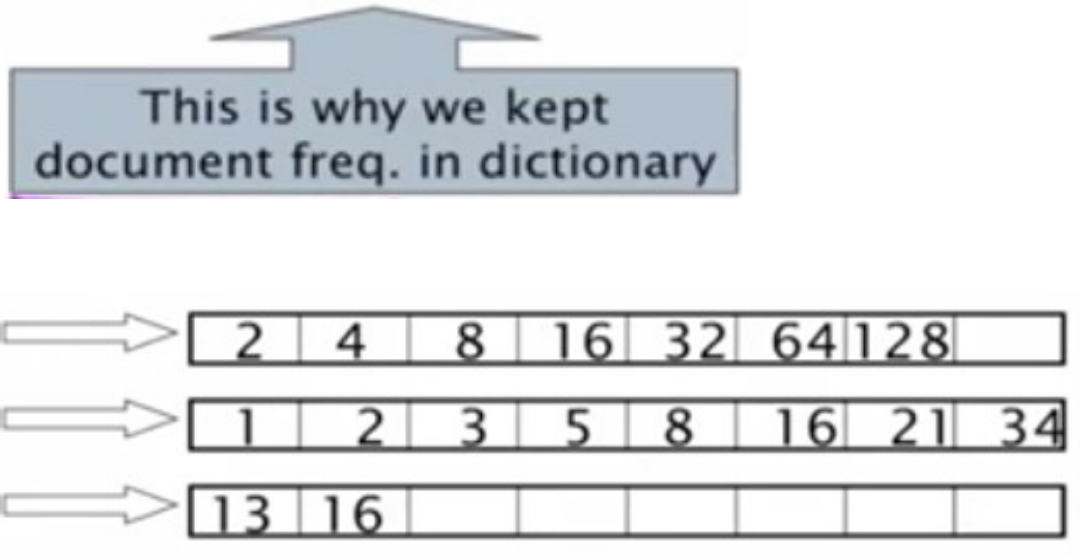
- What is the best order for query processing?
- Consider a query that is an And of n terms.
- For each of n terms, get its postings, then AND the together



- **Query**: **Brutus** AND **Calpurnia** AND **Caesar**

Query Optimization Example

- Process in order of increasing freq:
  - start with smallest set, then keep cutting further.



More general optimization

- e.g., (madding OR crowd) AND (ignoble OR strife)
- Get doc. For all terms.
- Estimate the size of each by the sum of its doc. freq.’s (conservative)
- Process in increasing order of OR sizes.
- OPTIMIZATION??

Exercise

Recommend a query processing order for

(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)

given the following postings list sizes:

Term	Postings size
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

A.

(i) (tangerine OR trees) = O(46653+316812) = O(363465)

(ii) (marmalade OR skies) = O(107913+271658) = O(379571)

(iii) (kaleidoscope OR eyes) = O(46653+87009) = O(300321)

Order of processing: a. Process (i), (ii), (iii) in any order as first 3 steps

(total time for these steps is O(363465+379571+300321) in any case)

B.

Merge (i) AND (iii) = (iv)

C.

Merge (iv) AND (ii): This is the only merging operation left.

Exercise

If the query is:

friends AND romans AND (NOT countrymen)

how could we use the frequency of countrymen in evaluating the best query evaluation order?

In particular, propose a way of handling negation in determining the order of query processing.