Information Retrieval
ITIS401
Chapter-5
2021-2022
Dr Mohamed Abdeldaiem

Lecture Overview

- **Models of Virtual Data Structure**
- **Concept of Models of Data**
- **Basic Data Elements and Structures**

  1-Scalar Variables and Constants

  2-Vector Variables

  3-Structures

  4-Arrays

  5-Tuples

  6-Relations

  7-Text

Models of Virtual Data Structure

- A model of data is a particular type of structure or manner of visualizing a data structure.
- One way of modeling gives consideration only to the constituent elements and sequencing or placement of data elements within other elements.
- A data structure is a collection of data elements or objects and relationships among them.
- These relationships concern the physical layout of the data objects or the semantic relations among them.

Concept of Models of Data

- A data structure begins with a single attribute and then is built up by combining attributes into larger elements.

Scalar Variables and Constants

- A variable is a data element that is a representation of an attribute and can take on differing values,i.e., whose value can vary.
- A scalar variable is a single instance of a variable. This is sometimes referred to as a field or item in computer programming terms.
- The variable is used to represent an attribute, and sometimes the words are used interchangeably.

Constants

- A constant, in computer terms, is the same as a variable, but its value may not change once the program using it has been compiled; hence, not during execution of the program.
- If we were using the value of in a calculation, we would store it as a constant (3.14159 . . .).

Vector Variables

- A vector is a variable that consists of a set of scalars, each representing an instance of the same attribute, such as a series of temperature readings for a hospital patient, or a series of subject headings descriptive of a book in a library catalog.

Structures

- The term structure can be used to denote a set of data elements, not necessarily all of the same attribute type.
- The term is often used in the narrow as well as in the broader sense in computer science.
- Context usually
- makes clear which is meant.

Arrays

- In mathematics, array simply means a rectangular arrangement of data. A vector is a one-dimensional array. A list of delinquent credit card numbers is one-dimensional.

Tuples

- The word tuple is a noun made out of a suffix, as in triple or quadruple, and which in turn comes from ply, meaning layer. A tuple is a one-layer structure.
- It represents one occurrence of a structure that may contain one or more vectors or other structures.

Relations

- A relation, in brief, is the set of all the tuples that exist for a given set of variables.
- In other words, a relation, in the database sense, empirically defines the content or semantic relationships among the variables constituting the tuples.

Text

- It is not clear how to classify a data element consisting of natural-language text, such as an abstract in a bibliographic record, the text of an article in a full text newspaper file, or even the response to a questionnaire item that adds, at the end of the list of choices: “Other (specify).” Text could be considered a scalar string variable of very great length.
- More modern systems will allow searching within a text variable for a particular sub string of characters, say the occurrence of STEROID within a text assumed to be dealing with athletics.

Text

- This is the most commonly used method in commercial database operations today.
- Finally, we could treat a text as a structure, made up of a series of words with a syntax relating them to each other or to the entity they describe.

Figure -1

Text

Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Words and position number in order of appearance

fourscore	1
and	2
seven	3
years	4
ago	5
our	6
fathers	7
brought	8
forth	9
on	10
this	11
continent	12
a	13
new	14
nation	15
conceived	16
in	17
liberty	18
and	19
dedicated	20
to	21
the	22
proposition	23
that	24
all	25
men	26
are	27
created	28
equal	29

Words and position numbers in alphabetic order

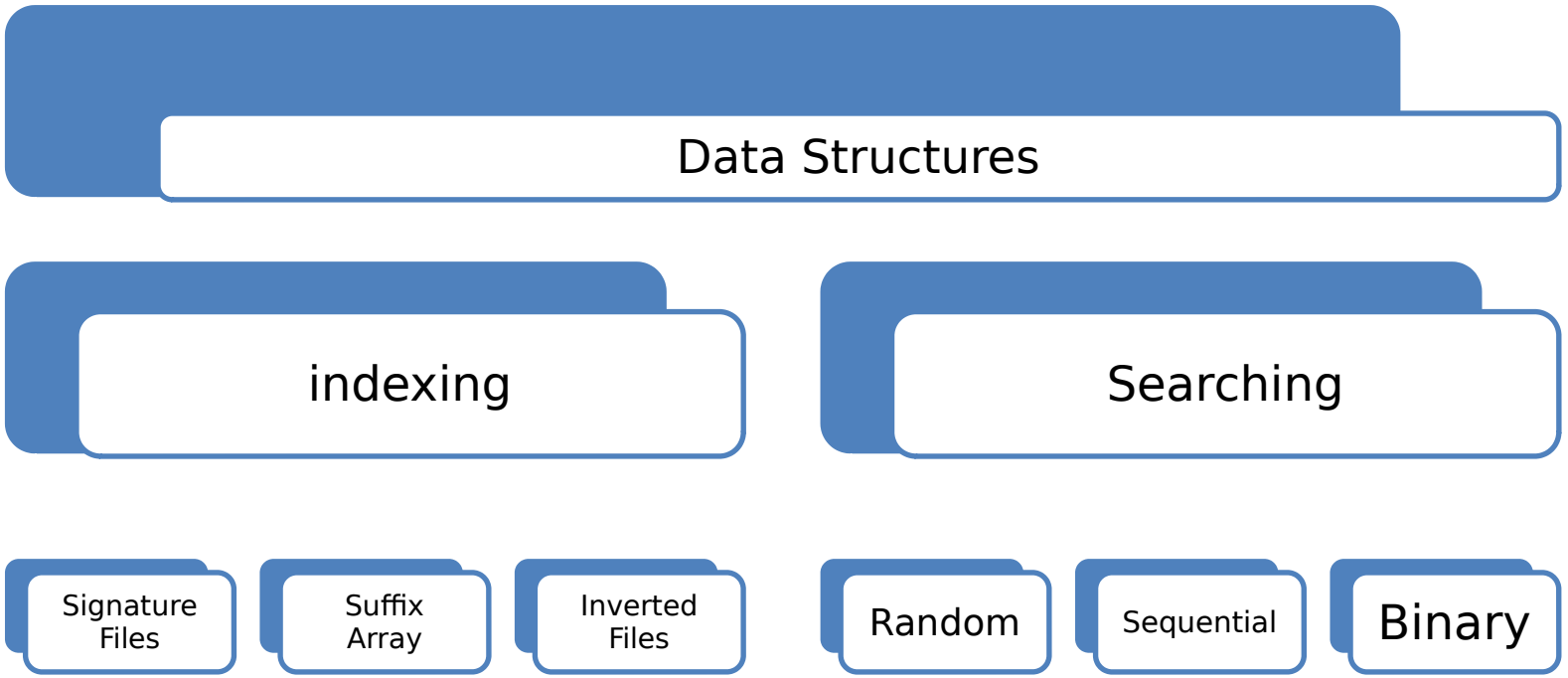
a	13
ago	5
all	25
and	2,19
are	27
brought	8
conceived	16
continent	12
created	28
dedicated	20
equal	29
fathers	7
forth	9
fourscore	1
in	17
liberty	18
men	26
nation	15
new	14
on	10
our	6
proposition	23
seven	3
that	24
the	22
this	11
to	21
years	4

An inverted file

- As shown are: a short text, the list of words in order of occurrence, with the sequential word number appended, and the same word list sorted into alphabetic order.
- The occurrence order of a word within a file enables a user to search for the phrase new nation rather than merely new and nation occurring anywhere with respect to each other because the location of the words can be seen to be adjacent and in the desired order.

Chapter-5
Part -2
How to build an inverted file

Indexing and Searching



Inverted Files (Inverted Indexing)

Inverted Index is a **Word-Oriented Mechanism** for Indexing a Text Collection in order to speedup the Searching Task.

The Inverted File structure is composed of two elements.

Inverted Files Structure

1- The Vocabulary :

Is the set of all different **Words** in the **T*ext***.

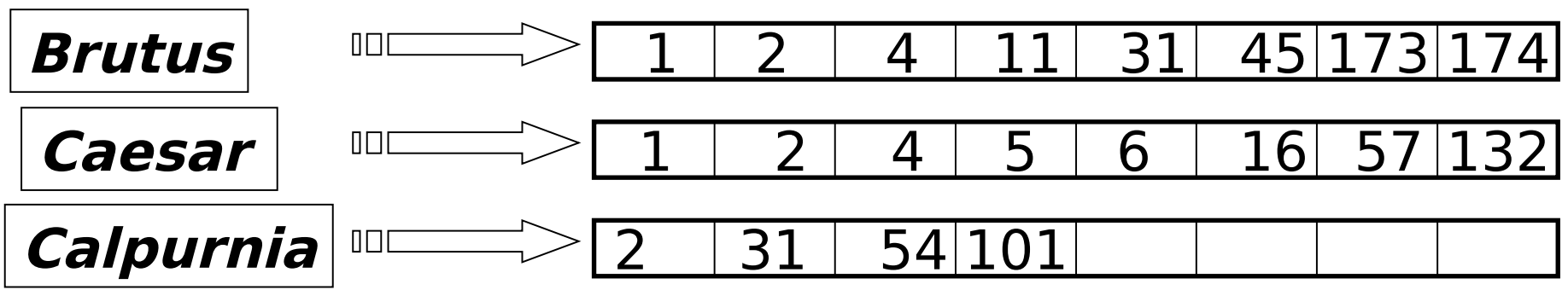
For each such **Word** a List of all the **Text** positions where the **Word** appears is **Stored.**

2- The Occurrences:

The set of those **Lists** is called the **Occurrences**.

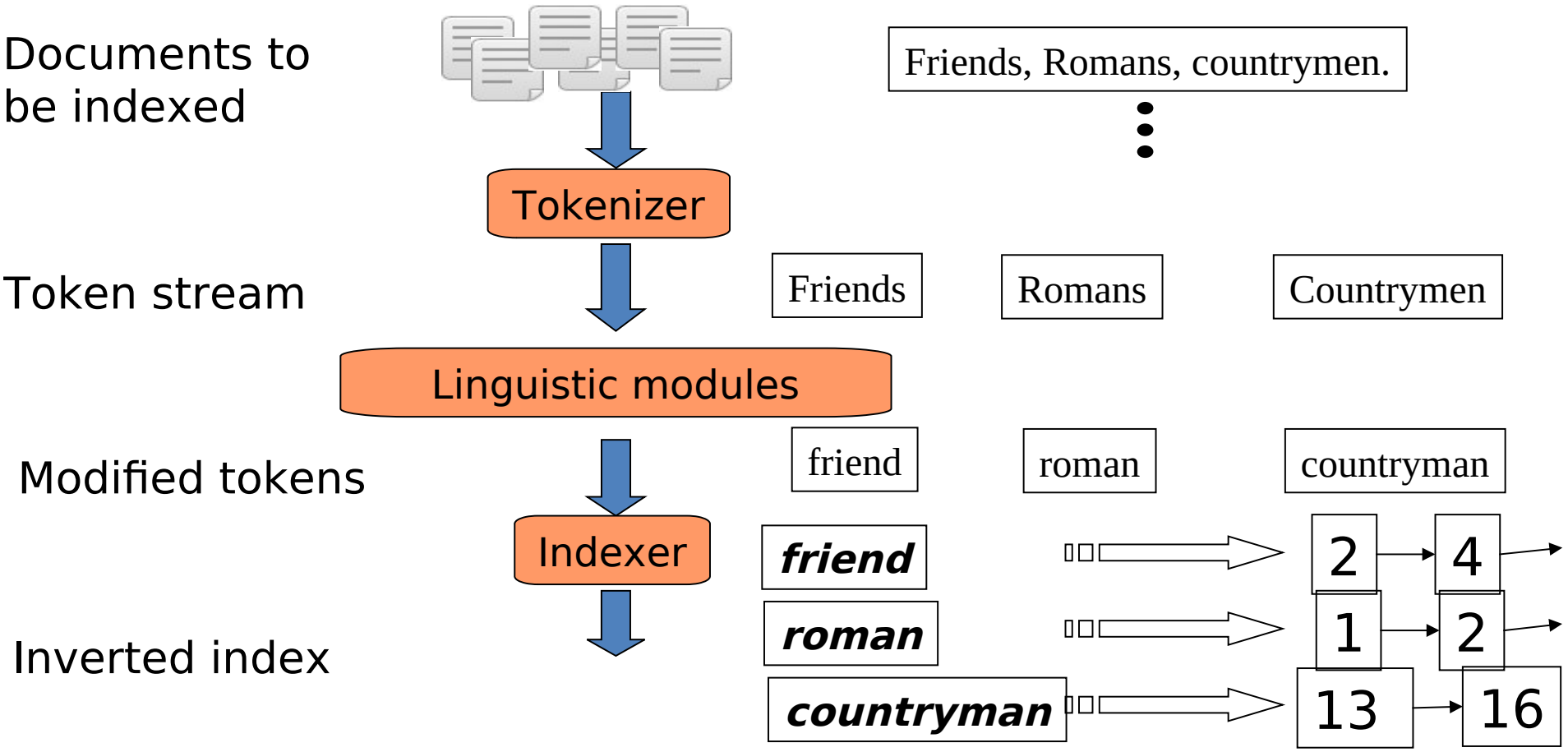
Inverted index

- For each term *t*, we must store a list of all documents that contain *t*.
  - Identify each doc by a doc-ID, a document serial number
- **Can we use fixed-size arrays for this?**



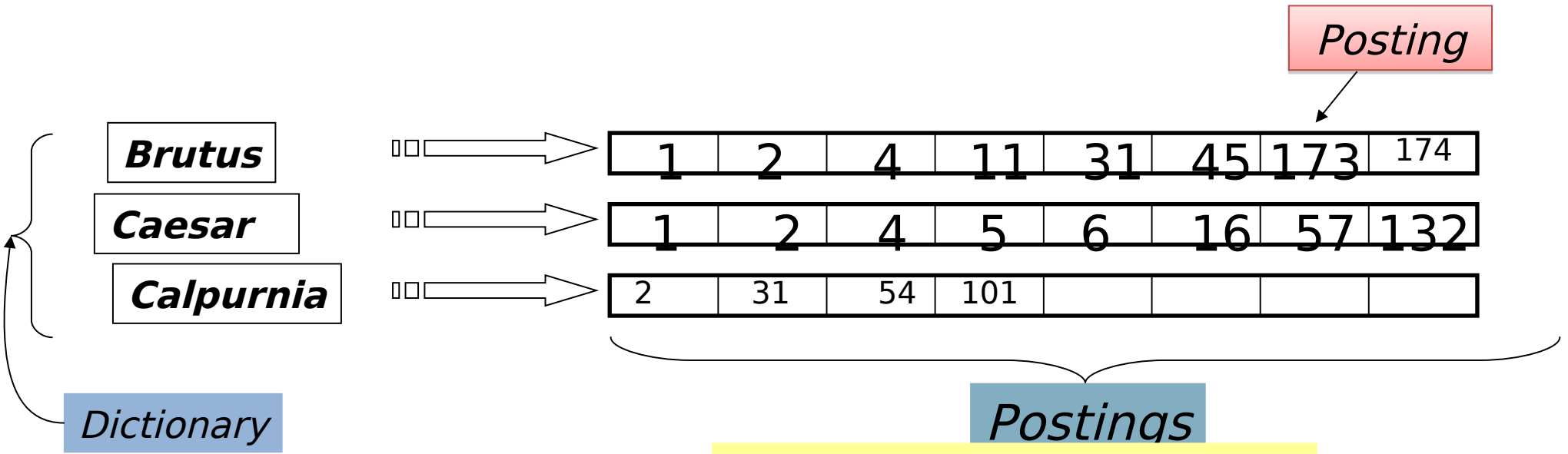
What happens if the word ***Caesar*** is added to document 14?

Inverted index construction



Inverted index

- We need variable-size postings lists
  - On disk, a continuous run of postings is normal and best
  - In memory, can use linked lists or variable length arrays
    - Some trade offs in size/ease of insertion



Initial stages of text processing

- **Tokenization**
  - **Cut character sequence into word tokens**
    - **Deal with *"John's"*, *a state-of-the-art solution***
- **Normalization**
  - **Map text and query term to same form**
    - **You want *U.S.A.* and *USA* to match**
- **Stemming**
  - **We may wish different forms of a root to match**
    - ***authorize*, *authorization***
- **Stop words**
  - **We may omit very common words (or not)**
    - ***the, a, to, of***

Indexer steps: Sort

- Sort by terms
  - And then doc-ID

Core indexing step

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Indexer steps: Token sequence

- Sequence of (Modified token, Document ID) pairs.

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

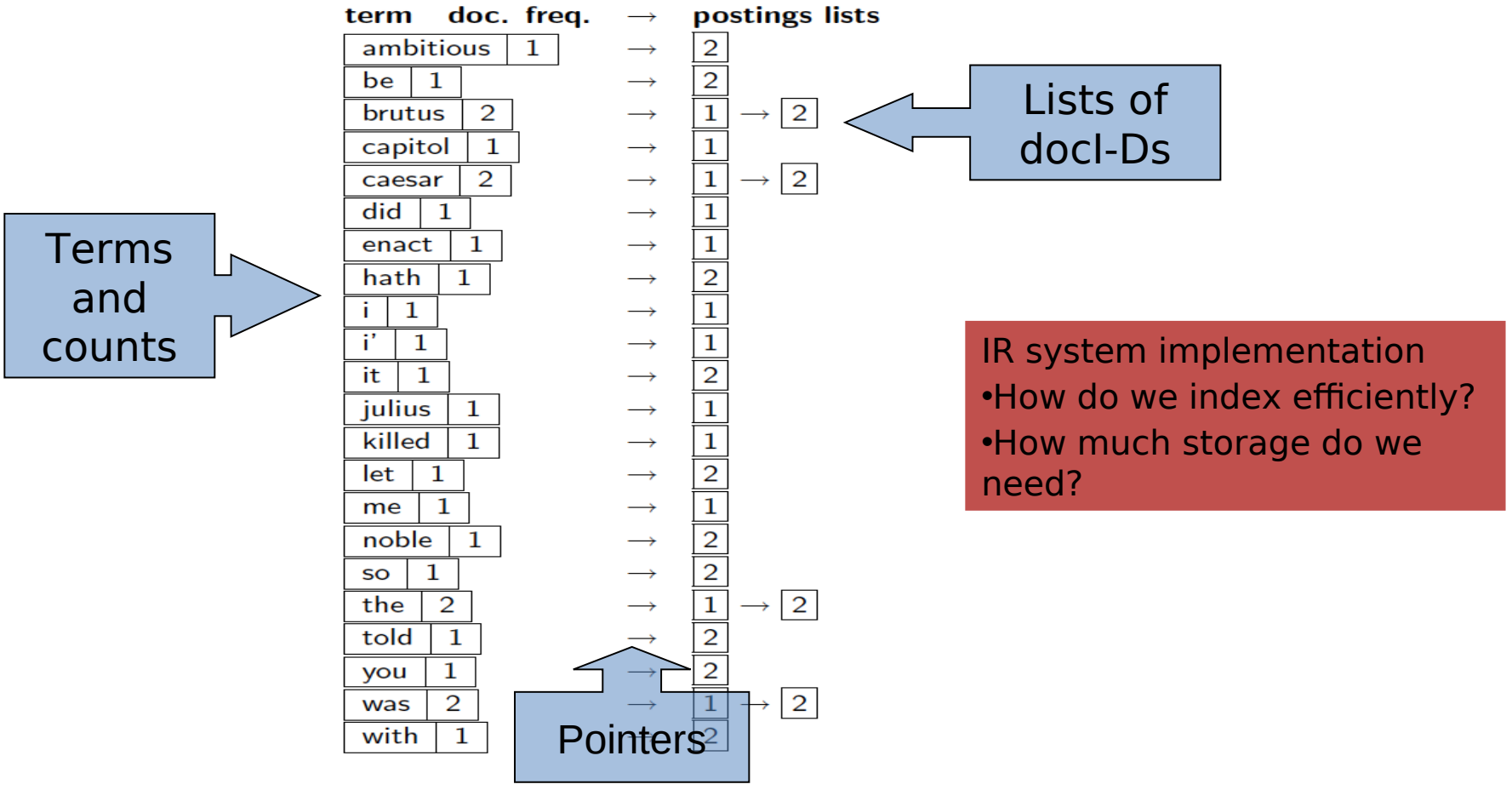
Indexer steps: Dictionary & Postings

- Multiple term entries in a single document are merged.
- Split into Dictionary and Postings
- Doc. frequency information is added.

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Where do we pay in storage?



End of Chapter-5

Any Question…?