INFORMATION RETRIEVAL

Unit 2 : Dictionaries and Query Processing

By

Rohini Naik

OUTLINE

- Components of Index
- Index Life Cycle
- Static Inverted Index
- Dictionaries-Types
- Index Construction

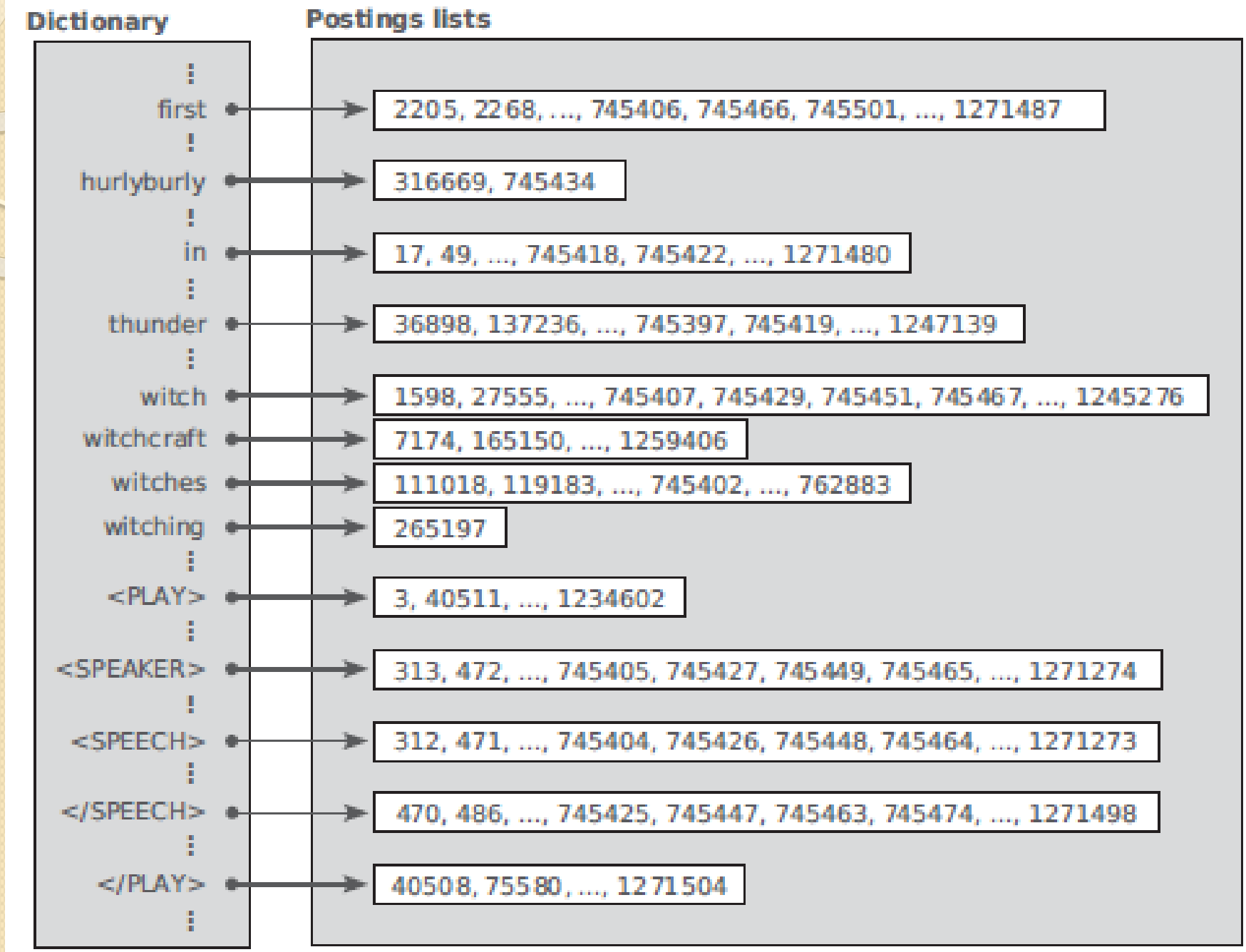
Components of Index

Dictionary: lists the terms contained in the vocabulary v

Posting lists: of the positions in which it appears

An inverted index is defined as an abstract data type (ADT) with four methods:

- first(t) returns the first position at which the term t occurs in the collection;
- last(t) returns the last position at which t occurs in the collection;
- next(t, current) returns the position of t's first occurrence after the current position;
- prev(t, current) returns the position of t's last occurrence before the current position.


Dictionary
Postings lists
first
2205, 2268, ..., 745406, 745466, 745501, ..., 1271487
hurlyburly
316669, 745434
in
17, 49, ..., 745418, 745422, ..., 1271480
thunder
36898, 137236, ..., 745397, 745419, ..., 1247139
witch
1598, 27555, ..., 745407, 745429, 745451, 745467, ..., 1245276
witchcraft
7174, 165150, ..., 1259406
witches
111018, 119183, ..., 745402, ..., 762883
witching
265197
<PLAY>
3, 40511, ..., 1234602
<SPEAKER>
313, 472, ..., 745405, 745427, 745449, 745465, ..., 1271274
<SPEECH>
312, 471, ..., 745404, 745426, 745448, 745464, ..., 1271273
</SPEECH>
470, 486, ..., 745425, 745447, 745463, 745474, ..., 1271498
</PLAY>
40508, 75580, ..., 1271504

- first(“hurlyburly”)=316669
- last(“thunder”) = 1247139
- first(“witching”) = 265197
- last(“witching”) = 265197
- next(“witch”,745429)=745451
  prev(“witch”, 745451) = 745429

Index Life Cycle

- Consists of two distinct phases :

1. Index construction: The text collection is processed sequentially, one token at a time, and a postings list is built for each term in the collection in an incremental fashion.

2. Query processing: The information stored in the index that was built in phase 1 is used to process search queries.

Dictionary

- The dictionary is the central data structure that is used to manage the set of terms found in a text collection
- It provides a mapping from the set of index terms to the locations of their postings lists
- At query time, locating the query terms' postings lists in the index is one of the first operations performed when processing an incoming keyword query.
- At indexing time obtain the memory address of the inverted list for each incoming term and to append a new posting at the end of that list.

- Dictionary implementations found in search engines usually support the following set of operations:

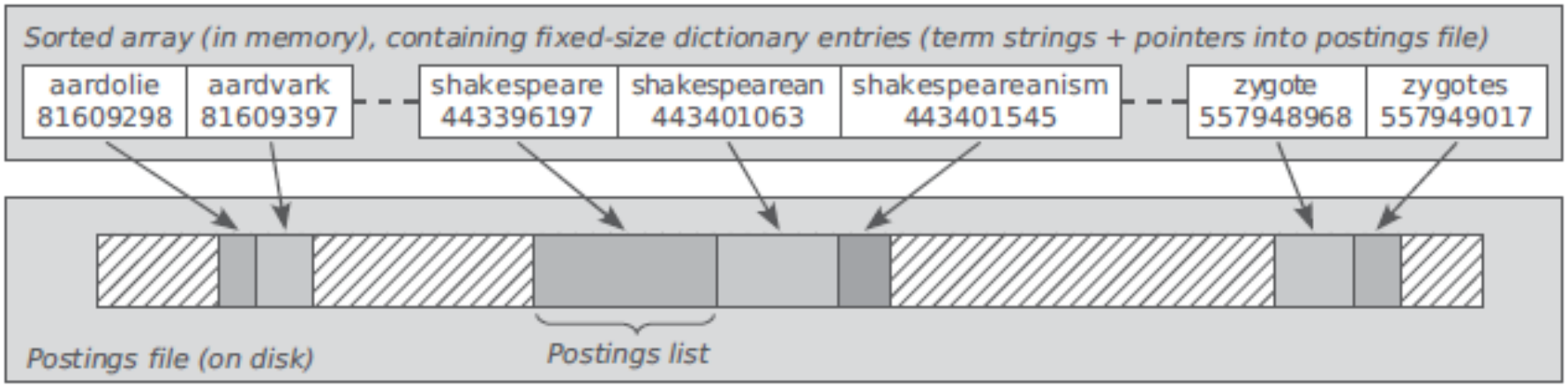
1. Insert a new entry for term T .

2. Find and return the entry for term T (if present).

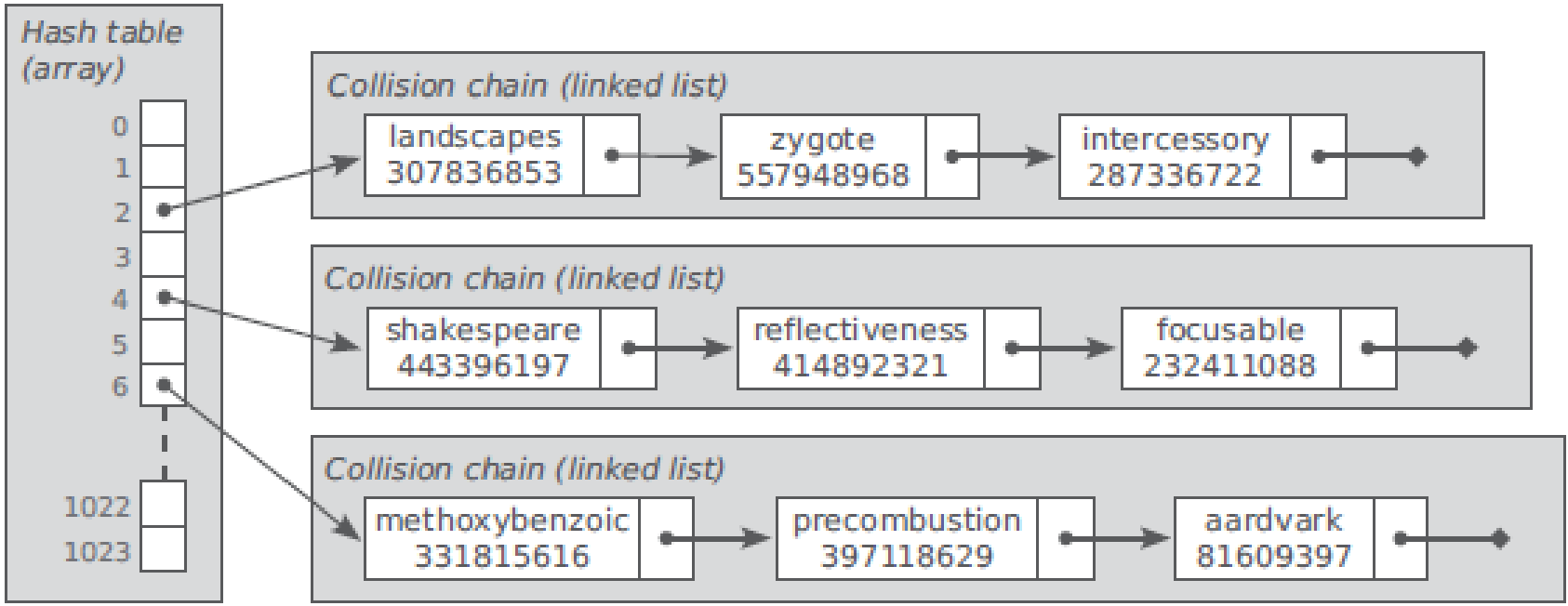
3. Find and return the entries for all terms that start with a given prefix P.

- The two most common ways to realize an in-memory dictionary are

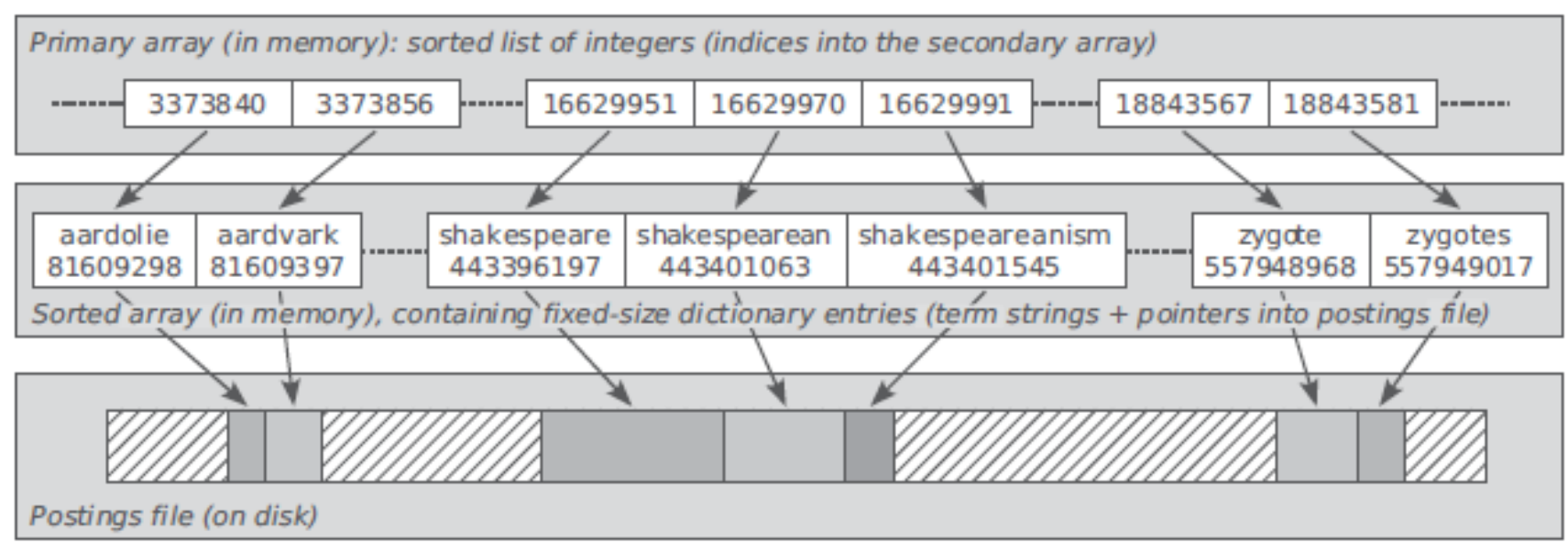
1. A **sort-based dictionary**, in which all terms that appear in the text collection are arranged in a sorted array or in a search tree, in lexicographical order . Lookup operations are realized through tree traversal (when using a search tree) or binary search (when using a sorted list).



2. A hash-based dictionary, in which each index term has a corresponding entry in a hash table. Collisions in the hash table (i.e., two terms are assigned the same hash value) are resolved by means of chaining — terms with the same hash value are arranged in a linked list.



Sort-based dictionary data structure with an additional level of indirection (the so-called dictionary-as-a-string approach).



Thank You!!