1 Introduction

Tries are data structures used in pattern matching. The lecture will start with standard tries and move on the space efficient compressed tries.

2 Text Processing

- We have seen that preprocessing the pattern speeds up pattern matching queries.
- After preprocessing the pattern in time proportional to pattern length, the KMP algorithm searches for an arbitrary English text in time proportional to the text length.
- If the text is large, immutable and searched for often, we may want to preprocess the text instead of the pattern in order to pattern matching queries in time proportional to the pattern length.

3 Standard Tries

- The standard trie for a set of strings S is an ordered tree such that:
  - each node but the root is labelled with a character.
  - the children of a node are alphabetically ordered from left to right.
  - the paths from the external node to the root yield the strings of S

Eg. S=bear, bell, bid, bull, buy, sell, stock, stop
4 Running time for operations

- A standard trie uses $O(W)$ space.
- Operations find, insert, delete take $O(dm)$ time, where:
  - $W =$ total size of strings in $S$
  - $m =$ size of string involved in operation
  - $d =$ alphabet size

5 Application of Tries

- A standard trie supports the following operations on a preprocessed text in time $O(m)$, where $m =$
  - word matching: find the first occurrence of word $X$ in the text
  - prefix matching: find the first occurrence of the longest prefix of word $X$ in the text
- Each operation is performed by tracing a path in the trie starting at the root.

6 Compressed Tries

- Trie with nodes of degree at least 2
- Obtained from standard trie by compressing chains of redundant nodes.
7 Why Compressed Tries

• A tree in which every node has at least 2 children has at most \( L-1 \) internal nodes, where \( L \) is the number of leaves.

• The number of nodes in a compressed trie is \( O(s) \), where \( s = \ldots \).

• The label in each node can be stored by using \( O(1) \) space index ranges at the nodes.

8 Tries and Web Search Engines

• The index of a search engine (collection of all searchable words) is stored into a compressed trie.

• Each leaf of the trie is associated with a word and has a list of pages (URLs) containing that word, called occurrence list.

• The trie is kept in internal memory.

• The occurrence lists are kept in external memory and are ranked by relevance.

• Boolean queries for sets of words (e.g., Java and coffee) correspond to set operations (e.g., intersection) on the occurrence lists.

• Additional information retrieval techniques are used.

References
