Algorithmic Problem Solving

APS2020 Strings I

Fredrik Heintz & Fredrik Präntare
Dept of Computer and Information Science
Linköping University

Suffixes and Prefixes

Suffix: A substring/affix occurring at the end of a word.

Prefix: A substring/affix occurring at the beginning of a word.

“banana” suffixes:

a, na, ana, nana, anana, (sometimes) banana

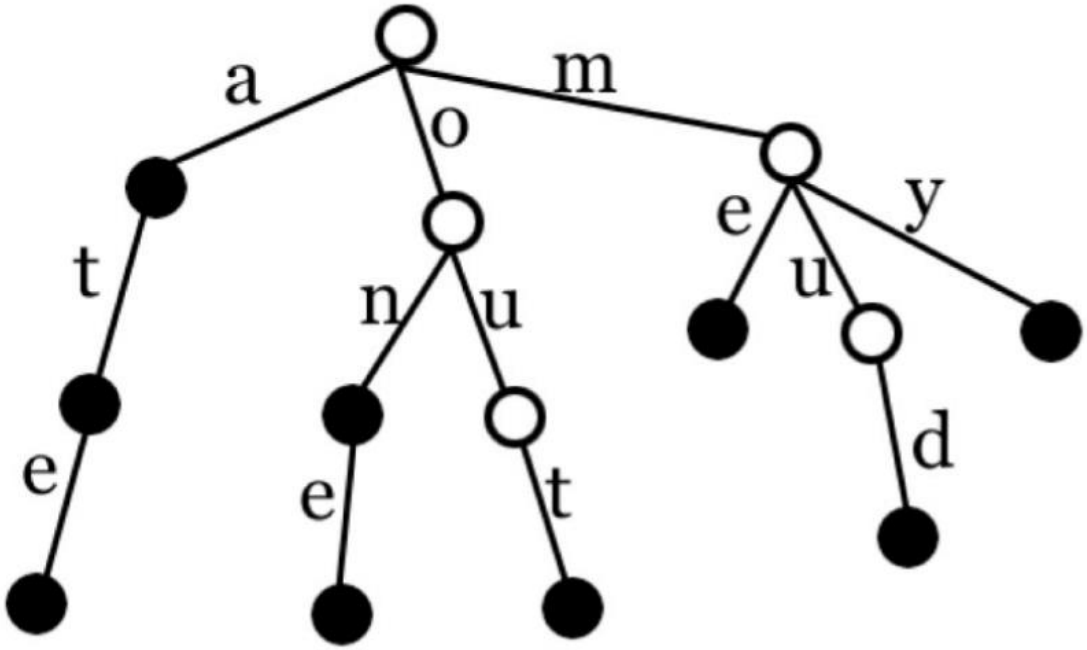
“banana” prefixes:

b, ba, ban, bana, banan, (sometimes) banana

Trie (Prefix Tree)

Trie: An ordered tree structure used for storing a set of data, usually strings, optimized for doing prefix searches

- Example: Does any word in the set start with the prefix `mart`?
- The idea: use a "26-ary" tree
  - each node has 26 children: one for each letter A-Z
  - add a word to the trie by following the appropriate child pointer



Trie (Prefix Tree)

Many applications:

- Dictionary
- Auto-complete
- Longest prefix matching
- Spell checking
- Substring problem

The String Matching Problem

Find all occurrences of a pattern ***P*** in a text ***S***.

The String Matching Problem

*Note that each substring **S**[i...j] is a prefix of the suffix **S**[i...n].*

Trie solution based on this observation:

1. Build a trie of all suffixes of ***S*** in *O(|S|^2)*.
2. Now *O(|**P**|)* for finding an occurrence of ***P*** in ***S***.

The trie (step 1) can be reused for future queries. Note that the time and memory complexity of this solution can be too hefty for many applications/problems.

The String Matching Problem

Solutions:

- **String library**
  C++ string::find, Java String.indexOf
- **KMP Knuth-Morris-Pratt *(lab 3.1)***
  O(|S|+|P|) time, O(|P|) space
- **Boyer-Moore**
  O(|S|+|P|) time, O(|P|) space,
  more efficient than KMP when the alphabet is large

The String Multimatching Problem

Find all occurrences of every pattern $P_1,\ldots,P_n$ in a text S.

The String Multimatching Problem

Solutions:

- **Aho-Corasick *(lab 3.2)***
  $O(|S| + |P_1| + \ldots |P_n| + k)$, where k is the total number of matches
- **Suffix Array**
  More on this next seminar!

Summary

- **This Week's Problems**
  - *Whac-a-Mole*
  - *(un)Fair Play*
  - *Jack and Jill*
  - *Towers of Powers 2*
- Trie
- The Substring & String Matching Problems