CS213/293 Data Structure and Algorithms 2024

Lecture 10: Trie: storing *string* → *Values*

Instructor: Ashutosh Gupta

IITB India

Compile date: 2024-09-09

If keys are strings!

The problem of storing maps boils down to storing keys in an organized manner.

- For unordered keys, we used hash tables
- For ordered keys, we used red-black trees.
- Let us suppose. Our keys are strings.

We have more structure over keys than total order. Can we exploit the structure?

Exercise 10.1

Can we define total order over strings?

Applications of string keys

- Web search
- All occurrences of a text
- Routing table (Keys are IP addresses)

Topic 10.1

Trie

Trie

Let letters of strings be from an alphabet Σ.

Definition 10.1

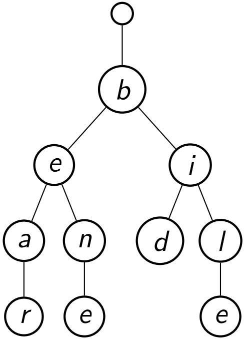
A trie is an ordered tree such that each node except the root is labeled with a letter in Σ and has at most $|\Sigma|$ children.

A trie may store a set of words.

A word stored in a trie is a path from the root to a leaf.

Example 10.1

In the following trie, we store words $\{bear, bile, bid, bent\}$.

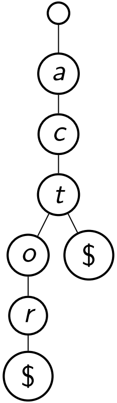


End marker

Sometimes a word is a prefix of another word. We need to add end markers in our trie. In our slides, We will use $.

Example 10.2

Consider set of words $\{act, actor\}$



Running times

- Storage $O(W)$, W is the total sum of lengths of words
- Find, insert, and delete will take $O(|\Sigma|m)$, where m is the length of the input word
  - At each node, we need to search among children for the node with the next letter.

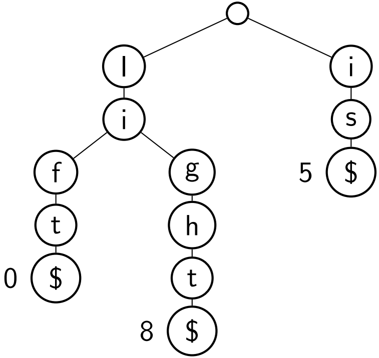
Application: word search

We may use trie to store the positions of all words of a text. The leaves of trie point at

- the first occurrence position of the word or
- a list of all occurrence positions.

Example 10.3

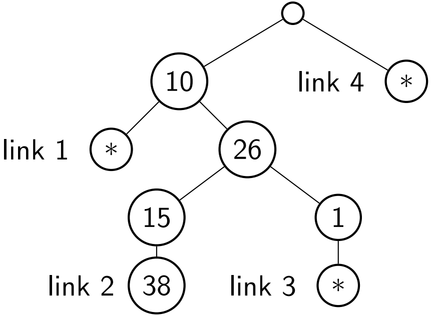
T	l	i	f	t		i	s		l	i	g	h	t



Application: routing table

Example 10.4

An internet router contains a routing table that maps IP addresses to links attached to the router.



The IP address of a packet is matched with the trie. The link with the longest match will receive the packet.

Exercise 10.2

Which link will receive the packets for following IP address?

- 21.10.1.6
- 10.26.10.6
- 10.26.1.6
- 10.26.15.9

Topic 10.2

Compressed trie

Compressed trie

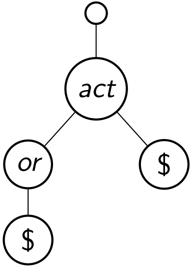
Definition 10.2

A compressed trie is a trie with nodes that do not have single children and nodes are labeled with substrings of words.

Obtained from standard trie by compressing chain of redundant nodes.

Example 10.5

In the following compressed trie, we store words $\{actor, act\}$.



Commentary: We did not prove exactly the following theorem. But, we can easily modify our earlier proof.

The number of nodes in the compressed trie

Theorem 10.1 (Recall)

If each internal node has at least two children, then the number of internal nodes is less than the number of leaves.

Each leaf represents a word.

Therefore, the number of internal nodes in the compressed trie is bounded by the number of words stored in the trie.

Does compression save the space?

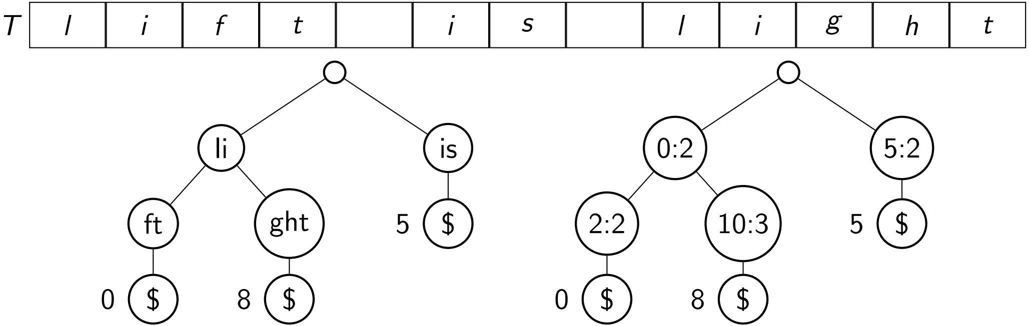
Typical usage of trie: fast search of words on a large text

The text must have been stored separately from the trie.

We need not store the strings on the nodes.

All we need to point at the position of the stored text and the length of the substring.

Example 10.6



T	l	i	f	t		i	s		l	i	g	h	t

Insertion and deletion on trie

Exercise 10.3

Give an algorithm for insertion and deletion in a compressed trie.

Topic 10.3

Suffix tree

Pattern search problem: Another perspective

Typical setting: We search in a (mostly) stable text T using many patterns several times.

Example 10.7

- *A text editor, where text changes slowly and searches are performed regularly.*
- *Searching in well-known large sequences like genomes.*

Can we construct a data structure from the text that allows fast search?

Suffix tree

Definition 10.3

A suffix tree of a text T is a trie built for suffixes of T.

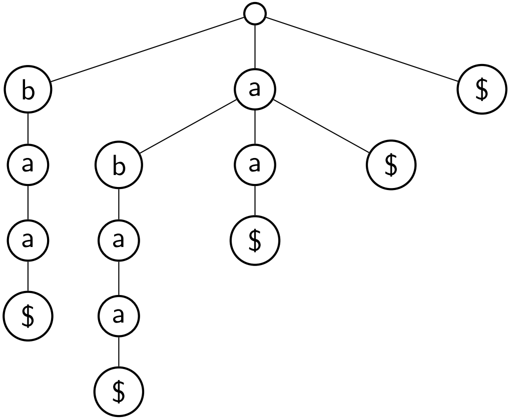
Exercise 10.4

How many leaves are possible for a suffix tree?

Example: suffix tree

Example 10.8

The following is the suffix tree of "abaa".



Usage of suffix tree

- Check if pattern P occurs in T.
- Check if P is a suffix of T.
- Count the number of occurrences of P in T.

Exercise 10.5

Using suffix tree find the longest string that repeats.

Suffix links

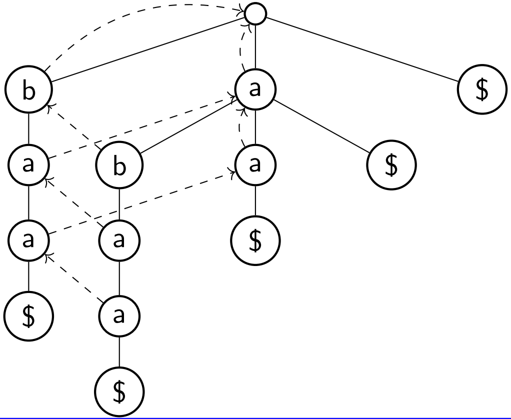
We can solve more interesting problems if we add more structure to our suffix tree.

Definition 10.4

In each node $x\alpha$, we add a pointer suffix link *that points to node α.*

Example 10.9

The following is the suffix tree of "abaa" with suffix links.



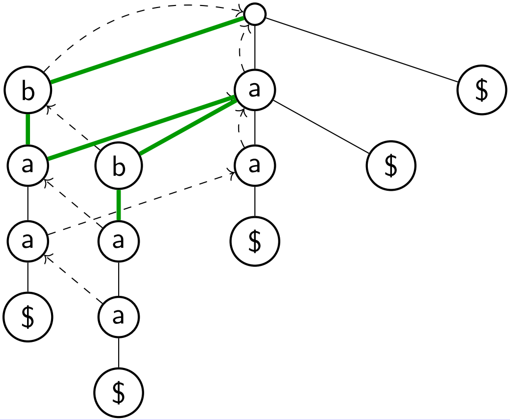
Usage of suffix links

Find the longest sub-string of T and P.

- Walk down the suffix tree following P.
- At a dead end, save the current depth and follow the suffix link from the current node.
- After exhausting P, return the longest substring found.

Example 10.10

The following is the suffix tree of $T =$ "abaa". Let us find the longest sub-string of "baba" and T.



Topic 10.4

Constructing suffix tree

Suffix tree construction

If we have the suffix tree for $T[0:i-1]$, we construct the suffix tree for $T[0:i]$.

In the order of the suffix links, Insert $T[i]$ at the end of each path of the tree,

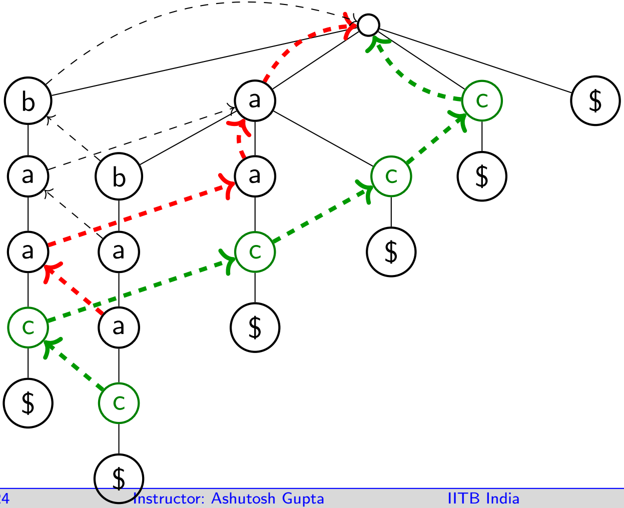
Exercise 10.6

a. What is the complexity of the above algorithm?

b. What will be the complexity of suffix tree construction without suffix links?

Example 10.11

Let us add c in the suffix tree of "abaa".



Ukkonen's algorithm

Suffix links give us $O(n^2)$ construction, can we do better?

Yes.

Ukkonen's algorithm uses more programming tricks to achieve $O(n)$. We will not cover the algorithm in this course.

Topic 10.5

Tutorial problems

Exercise: suffix tree

Exercise 10.7

Compute the suffix tree for abracadabra\$. Compress degree 1 nodes. Use substrings as edge labels. Put a square around nodes where a word ends. Use it to locate the occurrences of abr.

Exercise: worst-case suffix tree

Exercise 10.8

Review the argument that for a given text T, consisting of k words, the ordinary trie occupies space which is a constant multiple of $|T|$. How is it that the suffix tree for a text T is of size $O(|T|^2)$? Give a worst-case example.

Commentary: Milind notes

End of Lecture 10