OH TODAY: 1:15 pm - 2:15 pm

Emails

Violation of course policies

Suffix Tri...
$\$ guarantees that no suffix is a prefix of another suffix.

Each path from root to leaf corresponds to a suffix in $T$ and corresponding to each suffix there is a path from root to leaf in the trie.

Q. Is $\$\$ a substring of $T$?

Every substring is a prefix of some suffix. Starting from the root we follow the edges corresponding to the characters in $\$\$ and if we do not "fall off" the trie then $\$\$ is a substring of $T$. 
Q. Is $\gamma$ a suffix of $T$?

Do the same as before and answer
YES if we end up at a node
which has an outgoing edge labeled
$\gamma$.

Q. How many $\gamma$ occur as a substring of $T$?

Follow the path corre. to show up
if we fall off the trie.

ans <= 0

else

ans <= \# leaf nodes in the subtree
rooted at the node we ended on.

Q. Longest repeated substring of $T$, i.e.,
a subseq of T that appears more than one in T?

**Solution:** Deepest node with $\geq 2$ children.

How many nodes does a suffix tree have?

w Works an suffix $f T$.

**Suppose** $|T| = m$.

Total # characters in all suffixes $= \frac{m(m+1)}{2}$

$= \Theta(n^2)$.

Can I tests in which the # nodes in the tree is $O(m)$?

$T: a a a \ldots a$

$\underbrace{m}$
Compressed trie \( \rightarrow \) Suffix tree

- coalescing all edges from node

Class of trees in which each node is the

\[ \Theta(n^2) \]

\[ O(m^2) \]
with 1 into an edge.

# node n a suffix tree T text T with
# char = O(m).

T: abbababa

Hint

(3, 6)
at least as good as a full binary tree.

Building a suffix tree:
- Build the standard trie.
- Convert into compressed trie (suffix tree).

Suppose $|T| = m$, then:
-时间: $O(m^2)$
-空间: $O(m^2)$
Alg.: Start with the longest suffix & "break" the edges as you process smaller suffixes.
\[
T: a^n b^n \quad \equiv \quad a^3 b^3
\]

\[\text{ukkonen: } O(m) \text{ time } \& \ O(m) \text{ span}\]
online alg.

1. Pattern P: All indices whom pattern P occurs in T.
   - Look for the pattern in the tree.
   - Say you end up at node u.
   - Each leaf in the subtree rooted at u gives us the index in the text T whom the pattern

\[ |T| = n \] occurs.

Running time: \( O(n+k) \).

2. \( k \): # leaves in the tree rooted at the node whom we end up (node).

Q: Longest common substring?

\[ x = \text{tap} \quad y = \text{app} \]
Solution Consider the test formed by

\[ X \neq Y \]

- Tap \# app $\quad 01234567$

- Build a suffix tree for this test.

- Go to every node and annotate it as a \textit{X} node or a \textit{Y} node or an \textit{XY} node.

  - The rooted tree at this node has
  - \textit{leaves} indexed on the \textit{X} side and
  - \textit{\geq 1 leaf} indexed on the \textit{Y} side.
- Return the deepest XY node.