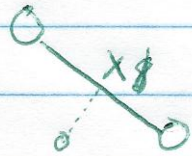
1 McCreight

Naive

T_i : $x[1 \ldots n+1] \ldots x[i \ldots n+1]$

T_1

T_2



n strings to add

n search time

$O(n^2)$

McCreight

Let $x[i \ldots n+1] = head(i)\,tail(i)$

Longest $LCP(i, j)$ $j < i$

Part already in the tree

How to find $head(i+1)$ from $head(i)$?

$head(i) = x[i \ldots i+h] \Rightarrow x[i+1 \ldots i+h]$ prefix of $head(i+1)$

Proof

$h = 0$

Trivial since $x[i+1 \ldots i+0]$ is the empty string ✓

$h > 0$

$head(i) = \boxed{a \;\; y}$

$\exists j : LCP(i, j) = ay$

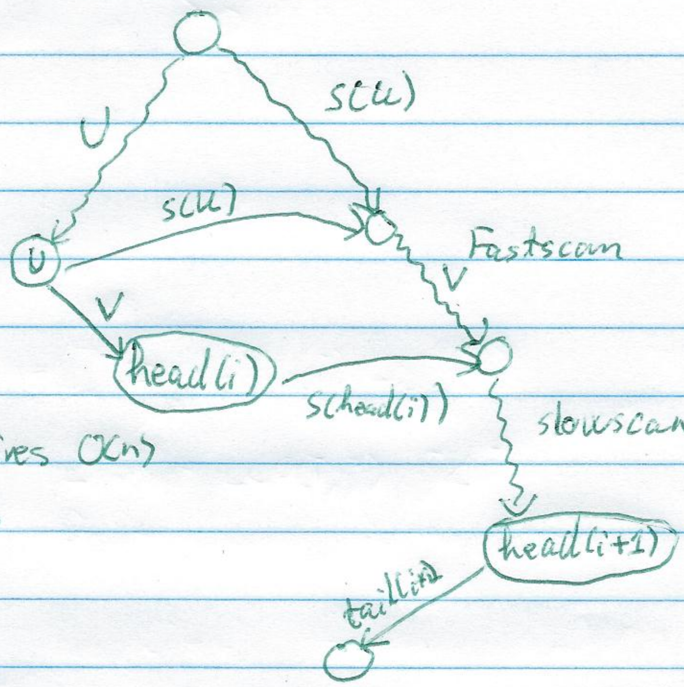
$\Rightarrow y$ prefix of $LCP(i+1, j+1)$

$\Rightarrow y$ prefix $head(i+1)$ ✓

How it works

T_i invariants

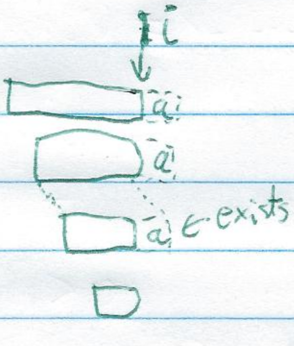
- non-terminal nodes have $s(-)$
- except maybe head(i)



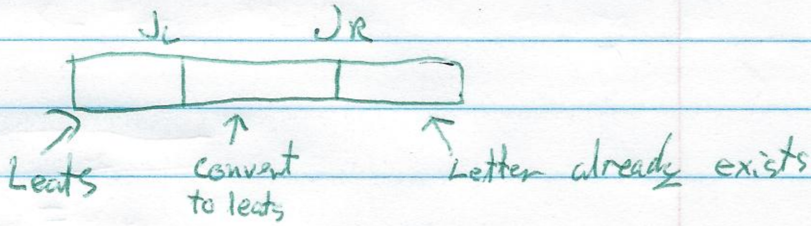
Optimization gives O(n) running time.

Ukkonen

T_i



Trick: if leaf represent as (j, ∞)
now free to add next letter

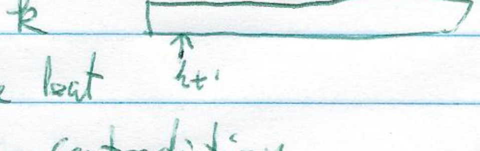


$j > 1$ leaf $\Rightarrow$ $j-1$ leaf

Proof

Antag $j > 1$ leaf og $j-1$ not leaf

$\exists k$:



$\Rightarrow j > 1$ ikke leaf

$\Rightarrow$ contradiction.