Applications of suffix tree

6.897
L18.4
Apr. 12, 2005

- count # occurrences of P: augment nodes with subtree size
- list first (say) k occurrences: $O(|P|+k)$
- longest repeated substring: deepest nonleaf node $O(|T|)$
  - ↳ in later depth
- multiple documents via multiple \$'s: $T = T_0\$_0 T_1\$_1 \cdots$
- longest common substring: deepest node with ② distinct \$'s below $O(|T|)$

Document retrieval [Muthukrishnan – SODA2002]

- extract **k** distinct documents containing P in $O(|P|+k)$ time
- P's subtree maps to interval $[i,j]$ of suffix array
- each $\$_k$ stores index of previous $\$_k$ (same document)
  - ↳ in suffix array
- want \$'s in interval $[i,j]$ with stored index $< i$
  - ⇒ not a repeat
- compute (position k of) min. stored index in $[i,j]$ via RMQ.
- if stored index at k is $< i$:
  - output it
  - recurse in $[i, k-1]$
  - recurse in $[k+1, j]$
- $O(1)$ time per output + $O(1)$ finale (per output)

LCA = longest prefix match of two substrings

- longest palindrome centered at T[i]
  - = longest common prefix of T[i:] & reverse(T)[|T|-i:]

⇒ longest palindrome substring in $O(|T|)$

- add-layers simple folds of maps [Arkin et al. – CGTA2004]