November 15, 2023 10:48 AM

STRINGS

Motivating Problem

Given "text" string $T = t_1 t_2 \cdots t_n \in \Sigma^*$,

build a static data structure s.t.

given "pattern" string $P = p_1 p_2 \cdots p_m$, $(m \leq n)$

decide whether P is a substring of T

(if so, report one occurrence or all occurrences or count)

eg. $T = 0\,1011\,011\,010$

$P = 1011$

Non-DS version: (Standard string matching problem)

trivial alg'm $O(mn)$ time

Knuth-Morris-Pratt '77 $O(n)$ time

(many other $O(n)$ algms...

eg. Karp-Rabin '87 in CS473)

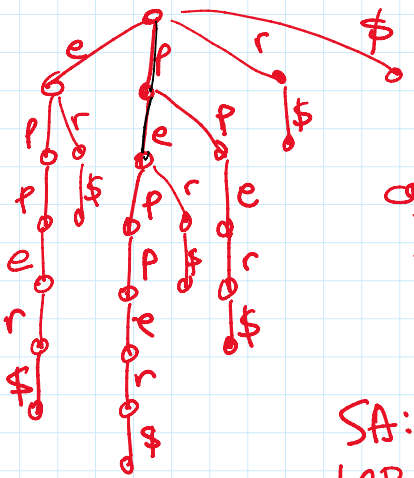
DS version?

Suffix Tree

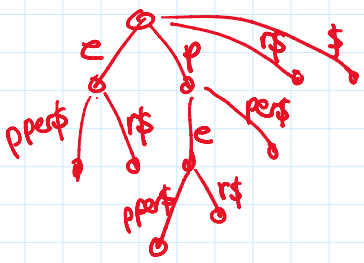
compressed trie for all suffixes of T.

eg. T = "pepper$"

1 pepper$
2 epper$
3 pper$
4 per$
5 er$
6 r$
7 $



compressed trie



O(n) size

P = "epp"
"pe"

SA: 2, 5, 1, 4, 3, 6, 7

LCP: 1, 0, 2, 1, 0, 0

query $O(m)$

or $O(m + \text{#occ})$.

or $O(m + occ)$.

Preproc time?? (algm for constructing suffix tree)

naive $O(n^2)$

Weiner '73 $O(n)$ for $|\Sigma| = O(1)$

complicated → $O(n \log n)$ for general Σ.

Farach-Colton '97 $O(n)$ for general $\Sigma \subseteq [n]$ (in word RAM)

Simplification: Suffix Array

Just store sorted order of all the suffixes of T

i.e. $SA[i]$ = index of i^{th} suffix in sorted order

also store $LCP[i]$ = length of longest common prefix between i^{th} & $(i+1)^{th}$ suffix in sorted order.

Obs suffix tree $\Longleftrightarrow$ suffix array + LCP

Pf: ($\Leftarrow$) repeated RMQ on LCP array. □

Algms to construct suffix array (+ LCP array):

naive $O((n \log n) \cdot n) = O(n^2 \log n)$

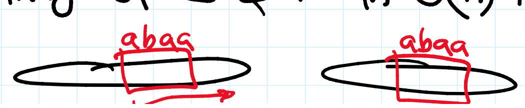
Gonnet et al. '92 / Manber-Myer '95 $O(n \log n)$

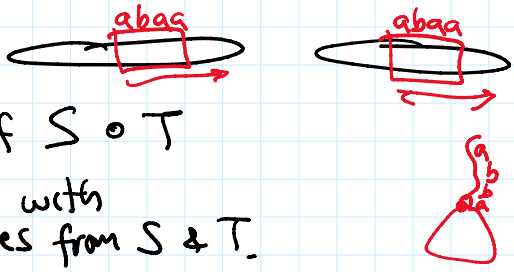
→ Kärkkäinen-Sanders '03 $O(n)$ simple!

⋮

Many many applications:

App 1 longest common substring of S & T in $O(n)$ time





Compute suffix tree of $S \circ T$

find deepest node with leaves from S & T.

App2 longest palindrome substring in S in $O(n)$ time

compute suffix tree of $S \circ S^R$

for each i, find LCA of i^{th} suffix & $(2n-i)^{th}$ suffix

Algm for Suffix Array by Kärkkäinen & Sanders:

idea - recursion

assume $\Sigma \subseteq [n]$.

Fact sort $[n] \times [n]$ or $[n] \times [n] \times [n]$ in radix sort in $O(n)$ time

1. let $T_1 = (t_1 t_2 t_3)(t_4 t_5 t_6)(t_7 t_8 t_9) \cdots$ (single char)

viewed as a string over Σ^3 of length $n/3$

can be mapped down to $[n/3]$ by replacing with ranks.

$T_2 = (t_2 t_3 t_4)(t_5 t_6 t_7)(t_8 t_9 t_{10}) \cdots$

$T_3 = (t_3 t_4 t_5)(t_6 t_7 t_8)(t_9 t_{10} t_{11}) \cdots$

2. recurse in $T_1 \circ T_2$

3. sort suffixes of T_3 by radix sort over $[n] \times [n]$ in

3. sort suffixes of T_3 by radix sort over $[n] \times [n]$ in $O(n)$ time.

how to compare

$(t_{3i+3}, t_{3i+4}\ t_{3i+5} \cdots) \overset{?}{<} (t_{3j+3}, t_{3j+4}\ t_{3j+5} \cdots)$

suffix in T_1 ; suffix in T_1

4. merge sorted list of suffixes of T_3 with suffixes of $T_1 \circ T_2$ in $O(n)$ time

how to compare

$(t_{3i+3}, t_{3i+4} t_{3i+5} \cdots) \overset{?}{<} (t_{3j+1}, t_{3j+2} t_{3j+3} \cdots)$

suffix in T_1 ; suffix in T_2

$(t_{3i+3}, t_{3i+4}, t_{3i+5} \cdots) < (t_{3j+2}, t_{3j+3}, t_{3j+4} \cdots)$

suffix in T_2 ; suffix in T_1

$$\Rightarrow \quad T(n) = T\left(\frac{2n}{3}\right) + O(n)$$

$$\Rightarrow \quad O\left(n + \frac{2n}{3} + \frac{4}{9}n + \cdots\right)$$

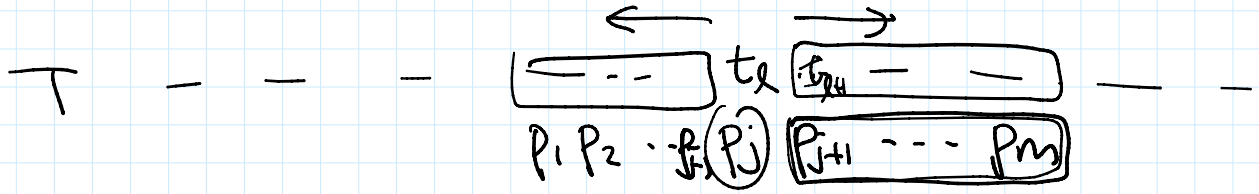
$$= \boxed{O(n)}$$

(can also compute LCP array ...)

More App 3: given query pattern $P = p_1 \cdots p_m$, decide whether $\exists$ match with ≤ 1 error

e.g. $T =$ "thisisastring"

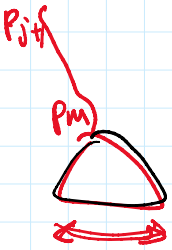
$P =$ "thin"



for $j = 1, \ldots, m$,

search suffix tree of T for $P_{j+1} \cdots P_m$ ←

$\Rightarrow$ interval $[a,b)$ in SA_T



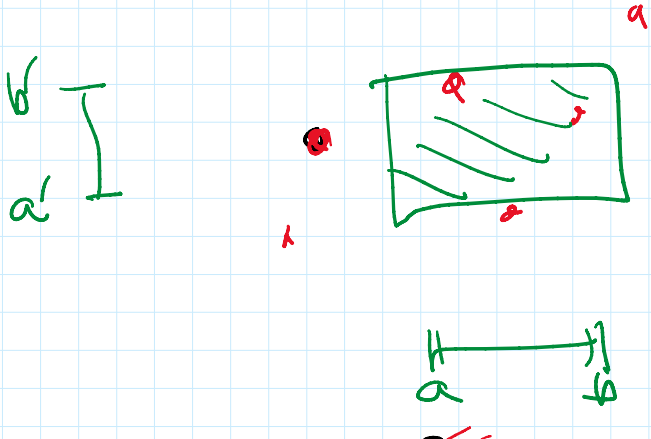
search suffix tree of T^R for $P_{j+1} \cdots P_1$ ←

$\Rightarrow$ interval $[a',b']$ in SA_{T^R}

find ℓ s.t. $x(\ell)$ position of $\ell+1$ in SA_T $\in [a,b)$

& $y(\ell)$ position of $n-\ell+1$ in SA_{T^R} $\in [a',b']$

$\Rightarrow$ find point $(x(\ell), y(\ell))$ in 2D rect.



$\Rightarrow$ 2D orth. range search

$\Downarrow$

2D range tree

$O(n)$ space

query time $O(\cancel{m^2}\ m + m \log n)$

(with suffix links)