**Strings**

**Motivating Problem**

Given a "text" string $T = t_1t_2...t_n \in \Sigma^*$

build a data structure s.t.

given a "pattern" string $P = p_1p_2...p_m$, $(m \leq n)$

decide whether $P$ is a substring of $T$.
(if so, report one occurrence or all occurrences or count)

e.g. $T = 0\boxed{10}1010$

$P = 1011$

**Non-DS version:**

trivial alg/m $O(mn)$ time

Knuth-Morris-Pratt '77 $O(n)$ time
(many other $O(n)$ algms...)

e.g. Kap-Rabin '87 in CS473

**DS version?**

**Suffix Tree**

compressed trie for all suffixes of $T$.

e.g. $T = "$peppers"

$P = "$epp" "$pe"

query $O(m)$ or $O(m+kocc)$. 

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

$LCP: 1,0,2,1,0,0$
Preproc time,?? (algm for constructing Suffix tree)
naive $O(n^2)$
Weiner '73 $O(n)$ for $|\Sigma|=O(1)$
$O(n \log n)$ for general $\Sigma$.
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. $SA[i] = \text{index of } i^{th}$ suffix in sorted order
Also store $LCP[i] = \text{length of longest common prefix}$
$\text{between } i^{th} \& (i+1)^{th} \text{ suffix}$
in sorted order.

Obs Suffix tree $\leftrightarrow$ suffix array + LCP
Pf: $(\subseteq)$ repeated RMQ. on LCP array.

Algs to construct suffix array ( + LCP array):
naive $O(n(\log n)^2)$ = $O(n^2 \log n)$
Gonnet et al. '92/Manneb-Myer '95 $O(n \log n)$
Kärkkäinen-Sanders '03 $O(n)$ simple!

Many many applications:
App1 largest common substring of $S$ & $T$ in $O(n)$ time
Compute suffix tree of $S \cdot T$
find deepest node with leaves from $S \cdot T$

___

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

compute suffix tree of $S \cdot S^r$
for each $i$, find LCA of $i$th suffix & $(2n-i)^{th}$ suffix

Algo for Suffix Array by Kärkkäinen & Sanders:

1. let $T_1 = (t_1 t_2 t_3) (t_4 t_5 t_6) (t_7 t_8 t_9) \cdots$
viewed as a string over $\Sigma^3$ of length $n/3$

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

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

2. recurse in $T_1 \cdot T_2$
3. sort suffixes of $T_3$ by radix sort over $[n] \times [n]$ in
3. Sort suffixes of T₃ by radix sort over \([n] \times \{\text{a}\}\) in \(O(n)\) time.

How to compare:

\[ (t_3, t_3i_3 t_3s_3 \ldots) \overset{?}{<} (t_3j_3 t_3j_4 t_3s_3 \ldots) \]

\text{suffix in } T_1,

\text{suffix in } T_1

4. Merge sorted list of suffixes of T₃ with suffixes of T₁ ∘ T₂ in \(O(n)\) time.

How to compare:

\[ (t_3j_3 t_3j_4 t_3s_3 \ldots) \overset{?}{<} (t_3j_4 t_3j_4 t_3j_4 \ldots) \]

\text{suffix in } T_2

\text{suffix in } T_2

\[ (t_3i_3 t_3i_4 t_3i_4 \ldots) \overset{?}{<} (t_3j_3 t_3j_3 t_3j_3 \ldots) \]

\text{suffix on } T_2

\text{suffix in } T_1

\(\implies\)

\[ T(n) = T\left(\frac{2n}{3}\right) + O(n) \]

\(\implies\)

\[ O(n + \frac{2n}{3} + \frac{4}{9}n + \ldots) \]

\[ = O(n) \]

(Can also compute LCP array \ldots)

---

More App 3: Given query pattern \(P = p_1 \ldots p_m\), decide whether \(T\) match with \(\leq 1\) error.

Eg. \(T = \text{"thisisastring"}\)

\(P = \text{"thin"}\)
for $j = 1, \ldots, m$,
search suffix tree of $T$ for $P_{i1}, \ldots, P_m$
$\Rightarrow$ interval $[a, b]$ in $SA_T$

search suffix tree of $T^R$ for $P_{i1}, \ldots, P_i$
$\Rightarrow$ interval $[a', b']$ in $SA_{T^R}$

find $x \in [a', b']$ s.t.
$\text{position of } P_i \in SA_{T^R} \cap [a, b]$
and $\text{position of } P_{i1} \in SA_T \cap [a, b']$

$\Rightarrow$ find point in 2D rect.

query time $O(m^2 + m \log n)$
(with suffix links)

$\Rightarrow$ 2D orth. range search

2D range tree
$O(n)$ space