6.3.17 L3

Topic: Strings

Terminology:

text T, $|T| = n$

pattern P, $|P| = m$

alphabet Σ

e.g binary, small (good), large (poly(n))

Pattern matching

Find occurence of P in T (as a substring)

There exist solutions in $O(n+m)$:

- Knuth, Mons, Pratt alg.
- Boyer-Moore

etc...

Idea: preprocess pattern

Data structure perspective:

preprocess T in $O(n)$

query P in $O(m)$

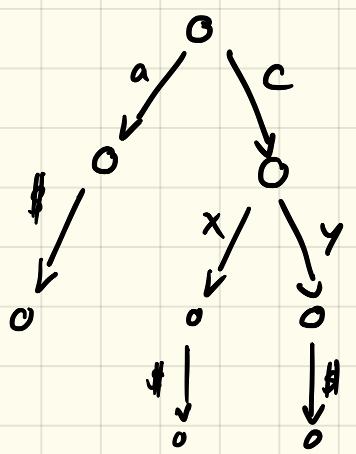
Problem 1

Preprocess $T_1, \ldots, T_k$ $\quad n = \sum_i |T_i|$

Query P $\quad$ is $P = T_i$,

or find pred./succ. of P in $\{T_i\}$ w.r.t. lex. order

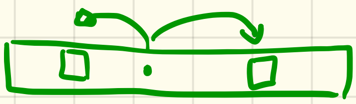
Trie: Rooted tree, where strings correspond to root-leaf paths.



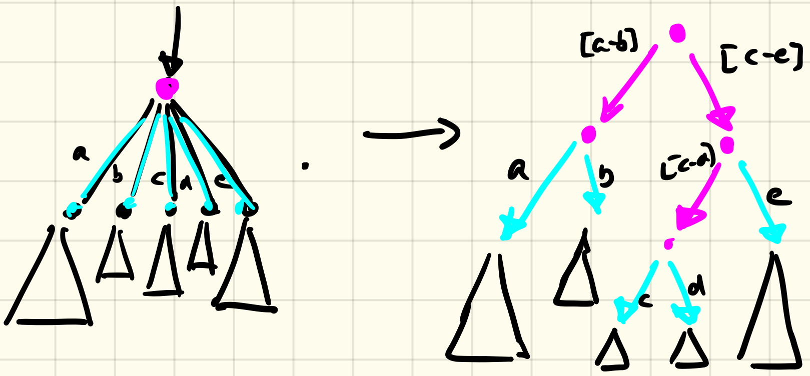
If the tree is in-order $\rightsquigarrow$ sorted T.

$T = \{a, cx, cy\}$

nodes store children	query	space
① array + blank cells store pointers [1]	$O(m)$	$O(n\lvert\Sigma\rvert)$
② balanced search tree (BST) (set/map)	$O(m \log\lvert\Sigma\rvert)$	$O(n)$
③ hash table [2]	$O(m)$	$O(n)$
③.5 van Emde-Boas tree	$O(m \lg\lg\lvert\Sigma\rvert)$	$O(n)$
③ + ③.5 hash + vEB [3]	$O(m + \lg\lg\lvert\Sigma\rvert)$	$O(n)$
④ weight balanced BST	$O(m + \log k)$ [4]	$O(n)$
⑤ indirection	$O(m + \lg\lvert\Sigma\rvert)$	$O(n)$

1 
2 no PRED/SUCC
3 works w.h.p. / vEB only used once
4 cf. next page

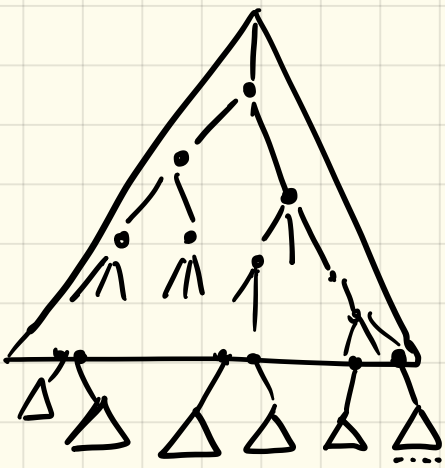
④



Claim: Going down twice:

- advances P
- reduces # candidates to $2/3$

⑤



① on branching nodes (BN) in top part ($\leq n/|\Sigma|$ BN)

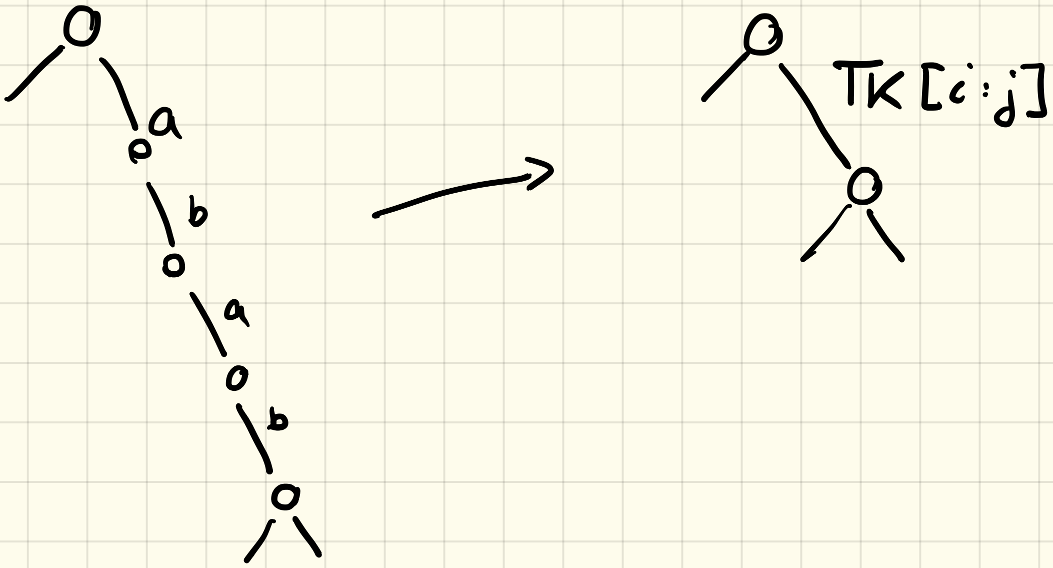
① on top leaves

② on non-branching

④ on bottom

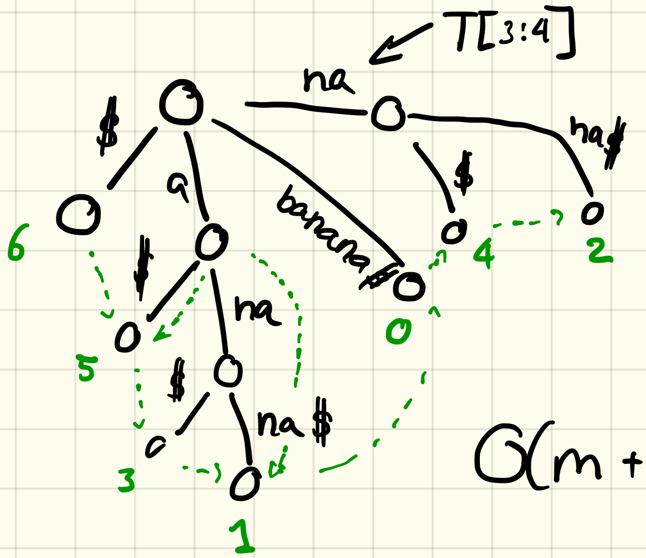
$\longrightarrow$ string sorting $O(n + k \log|\Sigma|)$

Compressed trie



Assume long input (e.g. war and piece)
and short queries

Suffix tree : compressed tie of all T[i:]



banana$

n+1 leaves

$O(n)$ space

$O(m + \lg|\Sigma| + k)$ all occurences of P

k = # of occurences

hashing: $O(m + k)$ all occ. of P

E.g. Longest Match of $T[i:]$, $[j:]$ = $LCA(i,j)$

→ How do we construct a suffix tree?

In $O(n)$ .... complicated ....

→ Consider instead <u>suffix array</u>.

SA		LCP
6	\$	
5	a\$	0
3	ana \$	1
1	anana \$	3
0	banana \$	0
4	na \$	0
2	nana \$	2

LCP: longest common prefix

~~$O(m \log n)$~~

$O(m + \log n)$

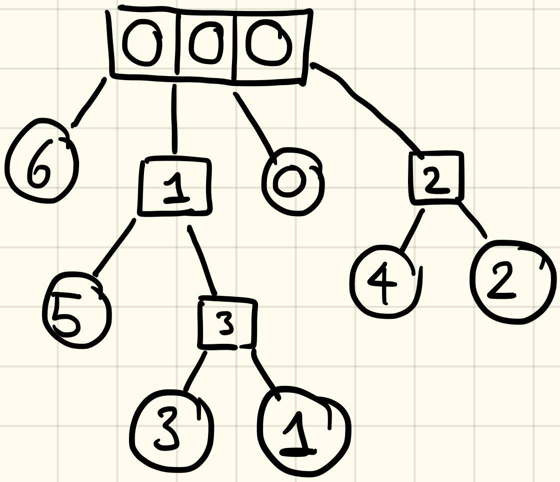
$LCP(T[i:], [j:]) = RMQ$ in of corresp. pos. in

<u>Claim</u> Construction of SA equiv. to ST.

ST → SA: in order traversal

→ euler tour gives LCP

SA → ST: Cartesian tree $\mathcal{O}(n)$ on LCP (mostly)



SA construction in $\mathcal{O}(n + \text{sort}(\Sigma))$:

1) Sort Σ,
2) replace Σ with $[1, \dots, |\Sigma|]$
3) $T_0 = [(T[3i], T[3i+1], T[3i+2])$ for $i = 0, \dots]$

   $T_1 = [$ " $3i+1$ " $+2$ " $+3$ " $]$

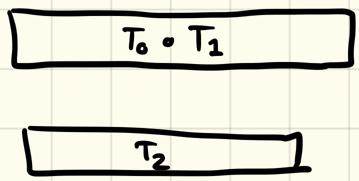
   $T_2 = [$ " $3i+2$ " $+3$ $+4$ " $]$
4) recurse on $T_0 \circ T_1$ = relative order of $T[3i:], T[3i+1:]$

5) radix sort of T_2

$$T_2[i:] \approx T[3i+2:] \approx (T[3i+2], T[3i+3])$$

$$\approx (T[3i+2], T_0[i+1:])$$

6) merge $T_0 \circ T_1$ with T_2



$T_0[i:]$ vs $T_2[j:]$

$\approx (T[3i], T_1[i:])$ vs $(T[3j+2], T_0[j+1])$

similarly T_2 vs T_1

e.g. banana\$

T_0 ban ana \$

T_1 ana na\$

T_2 nan a\$