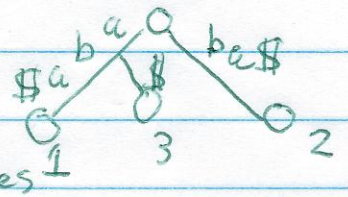
2 Suffix Array

Suffix Tree

$x = aba$



Folder meget $2n \cdot 36$ bytes

Suffix Array

$SA =$ | 3 | 1 | 2 |

~~Lexicalty~~ Lexicographical ordering

~~Can be constructe~~ Uses $|int| \cdot n$ space

Search

Suffix tree $O(m)$

Suffix array $O(m \cdot \log n)$ using binary search

Suffix tree replaced with suffix array without loss of time

Construction

using  $LCP(i, i-1)$

Can be constructed from a suffix tree using a depth first search in lexicographical order. $O(n \cdot a)$

Can be constructed directly in $O(n)$ time

mississippi
0 1 2 3 4 5 6 7 8 9 10

1) SA $i \bmod 3 \neq 0$

Use first 3 letters. Radix sort $O(n)$

iss
ssi
iss

Assign lex numbers. if all different we are done. If not construct

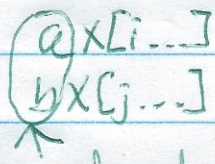
$u = \text{lex } i \bmod 3 = 1 \ \# \ \text{lex } i \bmod 3 = 2$

$SA(u)$ constructed with algorithm.

Can be mapped to needed SA.

2) SA $i \bmod 3 = 0$

Each string can be written as $aX[i\ldots]$ where $i \bmod 3 \neq 0$.



Sort on character if equal, sort using inverse SA from step 1

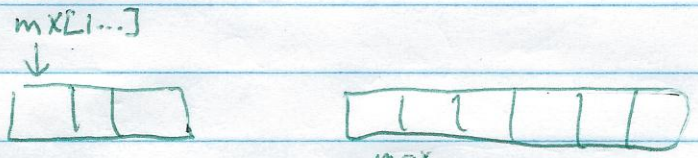
O(n) time.

3) Merge

Each string can be written as

$aX[i\ldots]$ $\quad i \bmod 3 \neq 0$

$abX[i\ldots]$ $\quad i \bmod 3 \neq 0$



Trick is to sort on first two characters and then use SA from step 1.

O(n) time.