Topic 8: KMP, AC Automata

Strings

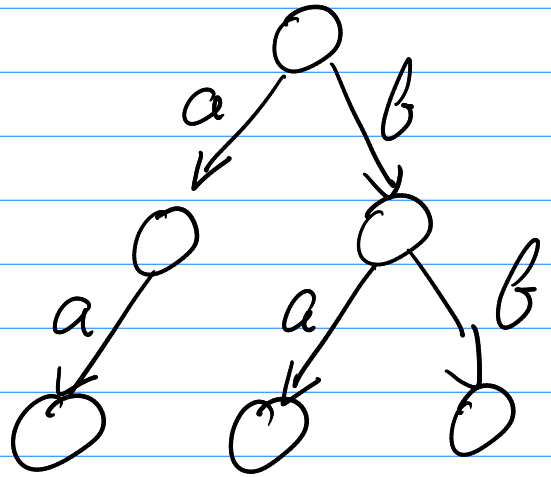
(See CS585 for theory)

- Rolling Hash (80% of the times)
- Automata (Trie, KMP, AC)

10% of the time

Automata

Trie is an automata

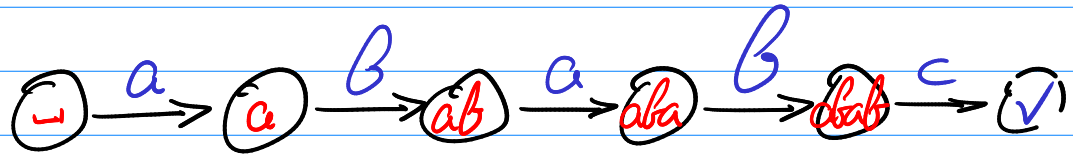


Tip of Your Tongue

KMP (String Matching)

$S = ababc$ $T = ab(ababc)$

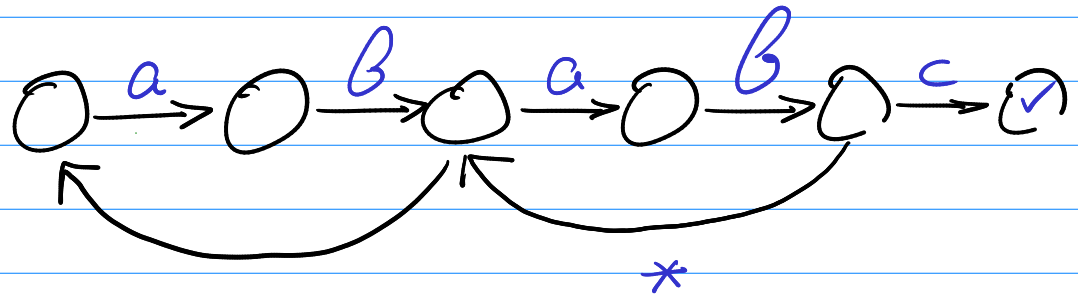
Automata-based approach



Question: What do we do if S does not match?

KMP (String Matching)

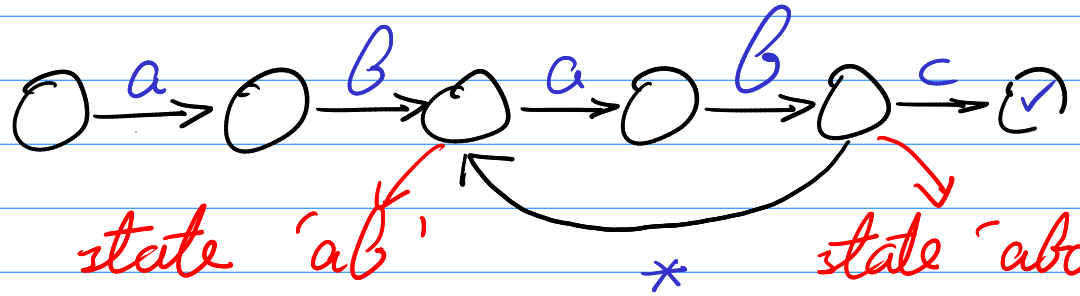
S = ababc T = abababc



Fail link: fall back to a previous state without consuming the character.

KMP (String Matching)

S = ababc T = abababc



Fail link: fall back to a previous state without consuming the character.

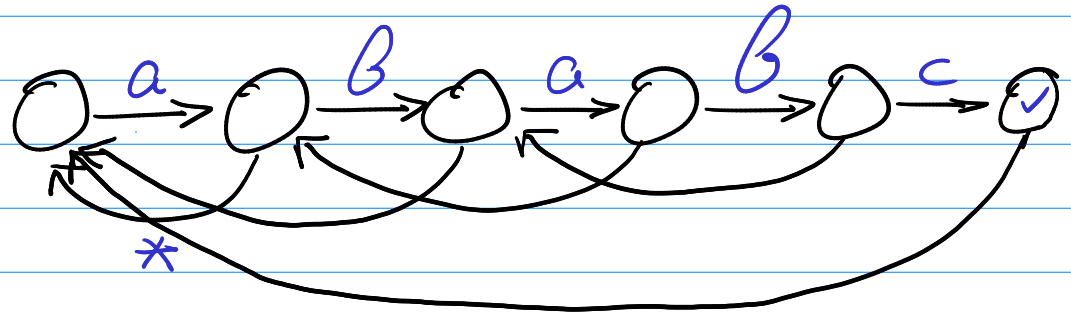
(Any string with suffix abab also has suffix ab.)

⇕

(state abab contains state ab as a suffix)

KMP (String Matching)

S = ababc T = abababc



Fail link: fall back to a previous state without consuming the character.

Compute fail['abab'] from fail['aba'] = 'a'

state abab ⟸ state aba

contains ab contains a

(as a suffix)

KMP (String Matching)

Compute fail['abab'] from fail['aba'] = 'a'

state i state abab ← s[i] state aba state (i-1)

contains cb ← s[j+1] contains a → state j

If state (i-1) contains state j as a suffix,
and s[i] = s[j+1], then state i contains
state (j+1) as a suffix.

State i contains fail[i], fail[fail[i]], ... as a suffixs.

```
k ← fail[i-1]
while k is not head
    if (s[k+1] = s[i]) break else k=fail[k]
if (s[k+1] = s[i]) fail[i] = k+1
else fail[i] = head.
```

KMP (String Matching)

```
k ← fail[i-1]
while k is not head
    if (s[k+1] = s[i]) break else k = fail[k]
if (s[k+1] = s[i]) fail[i] = k+1
else fail[i] = head.
```

Complexity?

Observe that whenever the while loop runs, fail must decrease.

fail only increase n times => O(n)

Constant Optimization: Compute all edges explicitly

go[i]['a'] ← go[fail[i]]['a'] (Slightly faster than falling back every time)

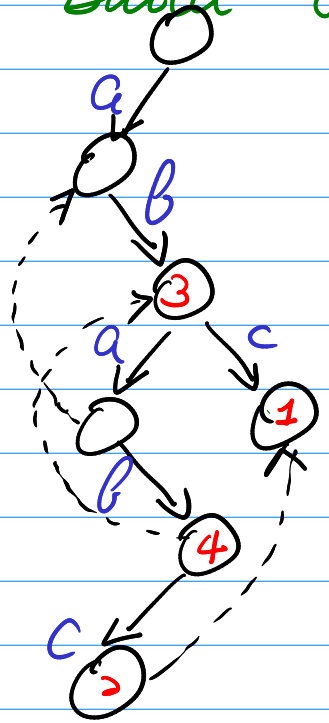
Compress Words

AC Automata (Multi String Matching)

$S_1 = abc$ $S_2 = abalc$ $T = ababafbc$

$S_3 = ab$ $S_4 = abab$

Very Easy : Build KMP on a trie !

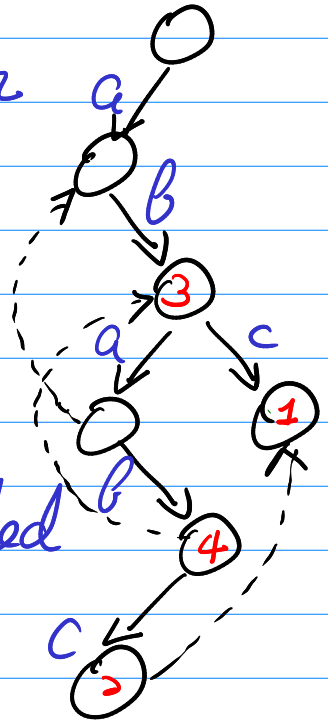


Intuitively every fail reduces height of the position so complexity stays the same.

AC Automata (Multi String Matching)

Caveat 1: One string may appear multiple times in the node.

e.g. 'ab' matches both on 3 and on 4.



In general, fail[i] (etc.) is also matched in position i.

Caveat 2: Large constant & TLE.

(Precompute go[i][char] if necessary)

Indie Album.