Let's begin at 9:05 PM

L98

Prefix Function and KMP Algorithm

Join Discord - https://bit.ly/ly-discord

RECAP

LearnYard

Problem is the same as the previous class

Given two strings `needle` and `haystack`, return the index of the first occurrence of `needle` in `haystack`, or `-1` if `needle` is not part of `haystack`.

LearnYard

Of course, there is a brute force way.

LearnYard

Of course, there is an efficient Rabin Karp way also.

LearnYard

But how to be a 100% sure that answer will be correct?

Prefix Function

Prefix fn. (π) of a string is array of length N where pi[i] represents the length of the longest propu prefix of s[0---i] which is also its suffix.

LearnYard

Eg. abcabcd $\Rightarrow$ [0, 0, 0, 1, 2, 3, 0]

"a" $\rightarrow$ 0

"ab" $\rightarrow$ 0

"abc" $\rightarrow$ 0

"abca" $\rightarrow$ 1

"abcab" $\rightarrow$ 2

"abcabc" $\rightarrow$ 3

"abcabcd" $\rightarrow$ 0

LearnYard

Brute Force

```
lps(n, 0)

for(i=0; i<n; ++i) {
    str = S[0 ---- i];
    for(l=1; l≤i; ++l)
        if (str[0--l-1] == str[i-l+1 --- i])
            lps[i] = l;
}
return lps;
```

$O(N^3)$ time

LearnYard

An observation

lps[i] can't be greater than lps[i-1] + 1

$S_0 \; S_1 \; S_2 \; S_3 \; - - - - \; S_{i-3} \; S_{i-2} \; S_{i-1} \; S_i$ 4

$$lps[i] \leq lps[i-1] + 1$$

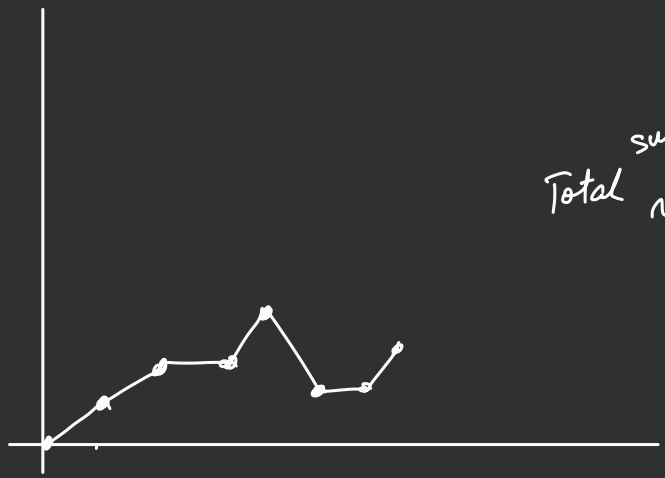
LearnYard

Optimisation 1

$O(N^2)$ time

```
for (i=1; i<n; ++i) {
    int pos = lps[i-1] + 1;
    string str = s[0---i]
    while (pos > 0 && s[0---pos-1] != s[i-pos+1--- i])
        pos--;
    lps[i] = pos;
}
```

LearnYard



Total sum of increments in lps value $\leq N$

$\Downarrow$

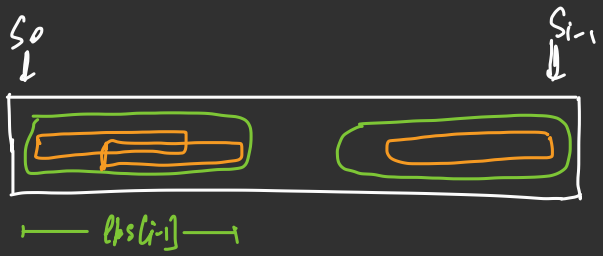
Total sum of decrements in lps value $\leq N$

LearnYard

Let's understand the final optimisation

$$\boxed{s_0\ s_1\ s_2\ s_3\ ----\ s_{i-2}\ s_{i-2}\ s_{i-1}}\ s_i$$

$$lps[i-1] \quad + 1$$



LearnYard

j^{i-1}

ababab efghagb ababab | a

$j = lps[i-1]$ // 6

$j = lps[j-1]$ $\Longrightarrow$ $ans[i] = 5$

LearnYard

Let’s implement this

LearnYard

2 Practice Problems
(Let’s try)

1. Pattern Find

LearnYard

Intuition / Solution

$\pi \Rightarrow$ pattern

str $\Rightarrow$ ab ab a b c d _ _ _ _ _ _

pat $\Rightarrow$ a b a b c d

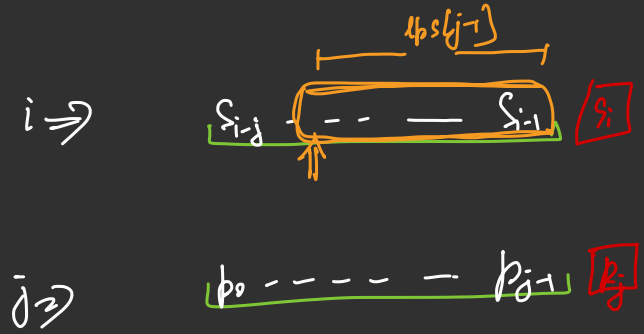
$i = 4$

$j = 4$

$s_0 \ldots s_{i-1}$ s_i

$p_0 \ldots p_{j-1}$ p_j

LearnYard




```
i = 0, j = 0,
while (i < l1) {
    if (s[i] == s[j])
        i++, j++;
    else if (j == 0)
        i++;
    else { // j != 0
        j = lps[j-1];
    }
}
```

LearnYard

Let’s implement

LearnYard

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!