Let's begin at 9:05 PM

L97
Naive Pattern Matching & Rabin Karp Algorithm

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

RECAP

What is the problem?

The Problem

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`.

haystack = "sad but happy" ⟹ 0

needle = "sad"

haystack = "sad but happy but"

needle = "but"

$\Longrightarrow$ 3

haystack = "sad but happy but"

needle = "butt"

$\Longrightarrow$ -1

Of course, there is a brute force way.

A better way? Think of some ideas.

A Possible Way

haystack ⇒ $h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8$ ----

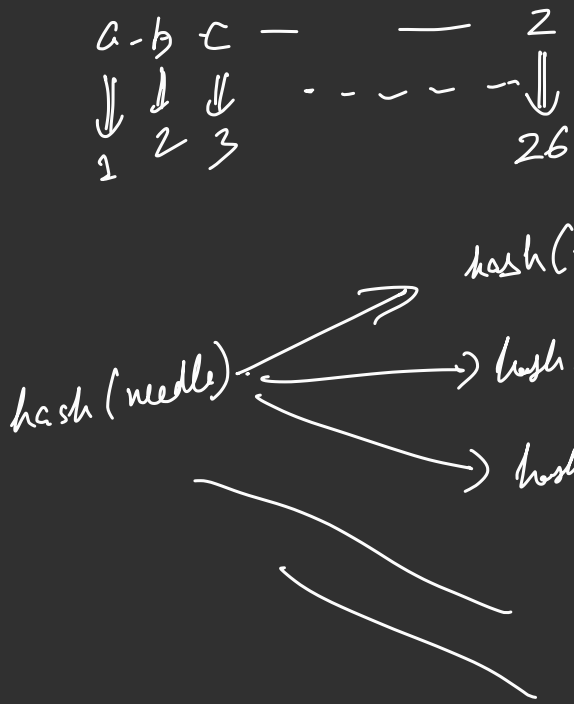
needle ⇒ $n_0\ n_1\ n_2\ n_3$

If $S_1 == S_2$, then $hash(S_1) == hash(S_2)$

hash() function is valid.

```
int hash(string s) {
    return 0;
}
```

⟹ Useless


a b c — — z
1 2 3 26
p = 31
hash(needle)
hash(h.substr(0, l-1))
hash(— (1, l))
hash(— (2, l+1))
1/1000

String Hashing

$h_1 \Longrightarrow \mod m_1$

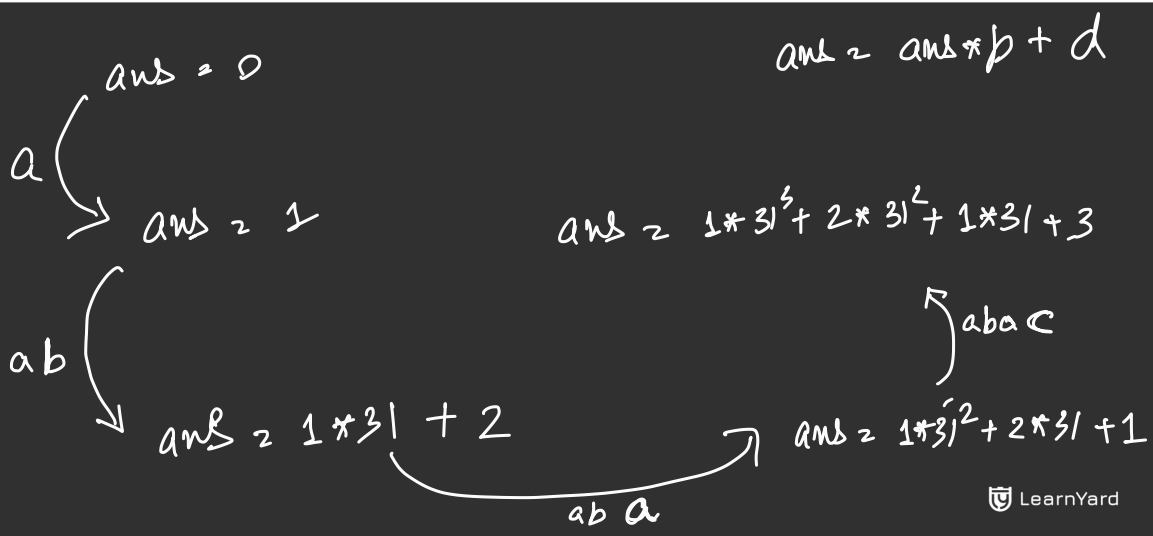
$h_2 \Longrightarrow \mod m_2$

$\frac{1}{10^6}$

$(hs_1 == ht_1 \;\&\&\; hs_2 == ht_2)$

$\hookrightarrow$ s is equal to t

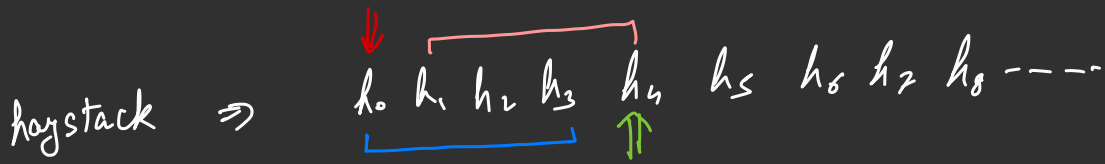
$a\ b\ a\ c \Rightarrow 1*31^3 + 2*31^2 + 1*31^1 + 3*31^0$



```
hash(S) {
    ans = 0;
    for(char c : S) {
        int d = c - 'a' + 1;
        ans = (ans * p + d) % m;
    }
    return ans;
}
```

$p \geq 31$ m

Rabin Karp Algorithm



$$h_{old} = d_0 * 31^3 + d_1 * 31^2 + d_2 * 31^1 + d_3 * 31^0$$

$$h_{new} = (h_{old} - d_0 * 31^3) * 31 + d_4$$

$$\hookrightarrow d_1 * 31^3 + d_2 * 31^2 + d_3 * 31^1 + d_4$$

needle ⇒ $n_0\ n_1\ n_2\ n_3$ ⟹ Calc. hash (needle) only once.

$$h_{new} = \left(h_{old} - dig_{del} * b^{lenNeedle-1} \right) * b + dig_{add}$$

Generic

Let’s implement this

1 Practice Problem

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!