20.538.979-2

orden

JORGE DE GOYENECHE
20.538.979-2

3)

a) 2)

d) Si poseemos el valor de $h(s_0 s_1 \ldots s_{p-1})$ podemos expresar el cálculo de $h(s_0 s_1 \ldots s_{p-1} s_p)$ de la siguiente manera

$$\sum_{i=0}^{p} \#(s_i) \mod 7$$

$$(a+b) \bmod c = ((a \bmod c) + (b \bmod c)) \bmod c$$

1

$$\left(\sum_{i=0}^{p-1} \#(s_i) + \#(S_p) \right) \mod 7$$

Por propiedades de mod:

$$\left(\left(\sum_{i=0}^{p-1} \#(s_i) \bmod 7 \right) + \left(\#(S_p) \bmod 7 \right) \right) \bmod 7$$

Donde el primer termino es $h(s_0 s_1 \ldots s_{p-1})$ por lo que toma $O(1)$ utilizarlo

$$\left(h_{p-1} + \#(s_p) \bmod 7 \right) \bmod 7$$

$\#(s_p)$ es conocido $\in \{1,24$
$\therefore O(1)$

Donde tenemos los costos:

$\#(s_p) \bmod 7 \quad O(1)$

$h_{p-1} + \#(s_p) \bmod 7 \quad O(1)$

$(h_{p-1} + \#(s_p) \bmod 7) \bmod 7 \quad O(1)$

Lo que asintoticamente nos da un costo de $O(1)$ hashear una combinacion de caracteres incrementalmente dado que conocemos el anterior.

$8 \bmod 7 \Rightarrow 1$

$5 \bmod 7 \to 5$

$7 \bmod 7 + 1 \bmod 7 =$
$0 + 1 \bmod 7 \to 1$

$$((\underbrace{2 \bmod 7}_{2}) + (\underbrace{3 \bmod 7}_{3})) \bmod 7$$

$2 + 3 \bmod 7 \to 5$

1 2 3 4 5 6 7 8 9 10

JORGE DE GOYENECHE

b) INDICES TABLA

$\sum \#(s) \bmod 7$

```
0 ———> (abbaa)
1      ∅
2      ∅
3 ———> (ab)
4 ———> (aab) -> (bb) -> (aab)
5 ———> (bab) -> (bab)
6      ∅
```

c)

TABLA ↑ T, TAMAÑO TABLA ↑ N, PALABRA ↗ w

```
1  PRINTPREFIJO (T, N, w):
2      aparecido = false
3      for i in |w|:
4          for j in N:
5              nodo = T[j]
6              while nodo:
7                  if w[0:i] = nodo.value[0:i]:
8                      print (nodo.value)
9                      aparecido = true
10                 nodo = nodo.next
11     if (!aparecido):
12         print(0)
```

```
struct nodo:
    string value;
    nodo* next;
```

|w| = tamaño palabra

w[0:i] = $w_0 w_1 \ldots w_i$

Imprime la palabra total de la cual w[0:i] es prefijo

Si lo interpreté mal, se imprimiría w[0:i] en su lugar, o una combinación para destacar qué prefijo se encontró, como print("w[0:i] -> nodo.value")