Algorithm Engineering
2/2/22

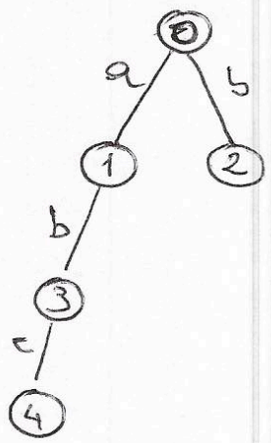
Q1.

T = a | b | a b a b c |

<0,0,a>
<0,0,b>
<2,4,<>>

LZ77

For LZ78 we build a Trie during the Text parsing:



a → <0, a>
b → <0, b>
ab → <1, b>
abc → <3, <>>

Q2 T = amata $

amata $
mata$a
ata$am
ta$ama
a$amaT
$amaTa

sort ⟹

L

$amaT	a
a$ama	T
amata	$
ata$a	m
mata$	a
ta$am	a

⟹ <atmaa, 2>

$L_0 = \{a, m, t\}$ → $L_1 = \{a, m, t\}$ → $L_2 = \{m, a, t\}$

atmaa

0atmaa

0tmaa

→ $L_3 = \{t, m, a\}$ → $L_4 = \{a, t, m\}$ → 02220

02aaa

02a2a

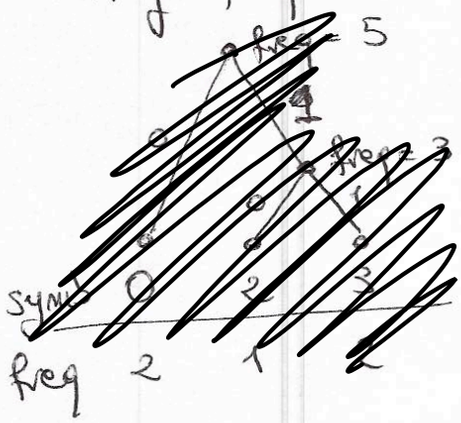
We increase by 1 the numbers ≥ 1.

0 3 3 3 0

Then we apply RLE0 with Wheeler's code

10 3 3 3 10 → 0 3 3 3 0

We finally compute Huffman Tree, for the symbols {0, 3} having frequencies: 2, 3 respectively. → code {0,1}



the string 0 3 3 3 0 is encoded as 0 1 1 1 0

Q3

l_{11} l_{12}

$l_1 = (1, 2, 4, 6, 9, 10, 15, 18, 20)$

$l_2 = (2, 3, 7, 8, 18)$

l_{21} l_{22}

$l_{11} \cap l_{21} = (1, 2, 4, 6)$

match $(2, 3)$ → l' l''

$l_{12} = (9, 10, 15, 18, 20)$

$l_{22} = (8, 18)$ → l'''

$l' \cap l'' = (4, 6)$

(3)

$l_{12} = (9, 10, 15, 18, 20)$

$l''' = (18)$ match

This means that $l_1 \cap l_2 = (2, 18)$

In the case of the two-level storage approach with block size $b = 3$

$L_1 = (\underbrace{1, 2, 4}, \underbrace{6, 9, 10}, \underbrace{15, 18, 20})$

L' $\underbrace{1}_{A_1}$ $\underbrace{6}_{A_2}$ $\underbrace{15}_{A_3}$ ← second level data structure

So we "merge" L' with $L_2 = (2, 3, 7, 8, 18)$ and thus we get the following distribution

$L_2 \cap A_1 = (1, 2, 4) \cap (2, 3) = 2$

$L_2 \cap A_2 = (6, 9, 10) \cap (7, 8) = \emptyset$

$L_2 \cap A_3 = (15, 18, 20) \cap \underbrace{(18)}_{} = 18$

these lists are obtained by merging $L' = (1, 6, 15)$ with L_2.

Q4.

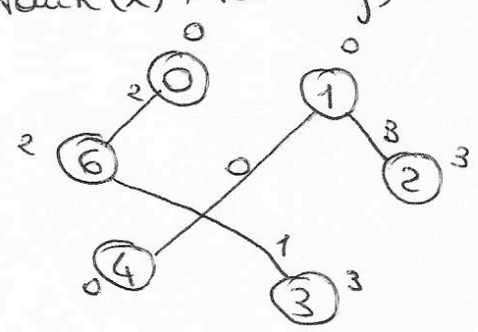
	h_1	h_2	h
aa	1	4	0
ac	3	6	1
bc	6	0	2
cc	2	1	3

rank	letter
2	a
3	b
4	c

$h_1(xy) = 3 \times \text{rank}(x) + \text{rank}(y) \mod 7$

$h_2(xy) = \text{rank}(x) + \text{rank}(y) \mod 7$

$h(t) = g(h_1(t)) + g(h_2(t)) \mod 4$



Q5

- Build the suffix array of $T \Rightarrow SA[1,n]$
- Search for P in SA and find the range $SA[l,r]$ of pattern occurrences.
- Since $SA[l,r]$ is not sorted by position, we sort it
- for $i = l$ to r :

  if $(i+q-1 \leq r)$ // $SA[i, i+q-1]$ is a subrange of $SA[l,r]$ of patter occurrences, namely they are q.

  and $(SA[i+q-1] - SA[i] \leq K)$ // the q-th occurrence and the first one of that raye have distance less than K.

  then print i;

print "-1";