(1) Harspool algorithm Time & Space trade off.

Pattern is shifted more than one.

In horspool Algorithm, a Shift table has to be Constructed which will Contain the no of shifts to the right.

EX: Consider BANGALORE — length of the String is 9

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
4	8	9	9	9	9	5	9	9	9	9	3	9	6	2	9	9	1	9	9

U	V	W	X	Y	Z	—
9	9	9	9	9	9	9

Last Character should not be consider.

ENGLISH Length - 7

E	N	G	L	I	S	Other Characters and -
6	5	4	3	2	1	7

KANNADA. Length - 7.

K - 6 in front of K How many Characters.

A - 2

N - 3

D - 1

Algorithm:

P → pattern
m – length of m
P [0..m-1]
ST[0 to asize-1] is the Shift table
asize is alphabet Size. [26 for A to Z]

INPUT : Pattern String P
OUTPUT : Shift table

```
Algorithm Shift_table( P, ST )
{
    for( i = 0 to asize-1)
        ST[i] = m;
    for( j = 0 to m-2)
    {
        index = Ascii value of P[j] – Ascii Value of 'A'
        ST[index] = m-1-j;
    }
}
```

asize = 26
RABBITS m = 8

when j = 0 index = Ascii Value of
R(P[0]) – Ascii value of A
82 – 65 = 17
ST[17] = 8-1-0 = 7

j = 1
index = Ascii value A[P[1]] – Ascii value of A
65 – 65 = 0.
ST[0] = 8-1-1 = 6
Similarly of other values

Case 1 :

T : - - - - - - K - - - - - - - Text Size = n
P : B A R B E R
B A R B E R Pattern = m

Compare last character if there is a mismatch then Shift the pattern m times

Case ii)

T : - - - - - - - - - - B - -
B A R B E R
B A R B E R

Shift for B is 2.

Case : (iii)

T : - - - - - - E R - - - - -
L E A D E R
L E A D E R

Shift Size should be size of the pattern and last Character is not in remaing Pattern.

Case : IV

T : - - - - - - X E R - -
B A R B E R - -
B A R B E R.

Now Shift Size should be 3 [if that last Character is occuring in m-1 Characters]

It depends on the right occurance of the last Character.

Shift value

$$t(c) = \begin{cases} \text{the pattern length } m \\ \text{if } c \text{ is not among the } n-1 \text{ characters of} \\ \text{the pattern} \\ \text{the distance from the rightmost } c \text{ among} \\ \text{the } n-1 \text{ characters of the pattern to} \\ \text{its last character; otherwise.} \end{cases}$$

BARBER - len - 6.

B - 2 other char - 6.
A - 4
R - 3
E - 1

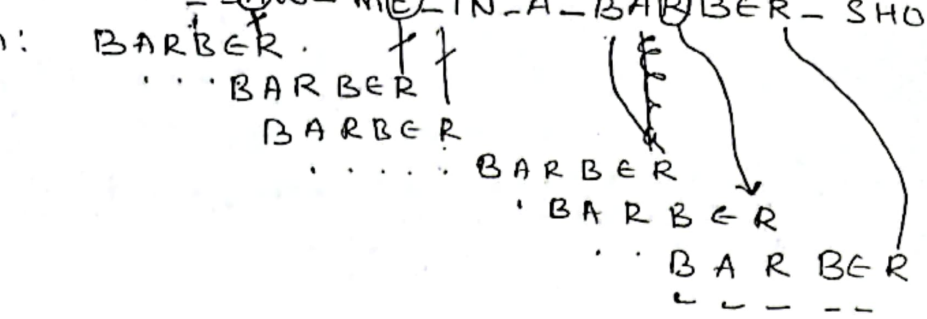
Pattern as LEADER

L - 5 other character - 6
E - 1
A - 3
D - 2
& -
R -

Ex°

TEXT : JIM_SAW_ME_IN_A_BARBER_SHOP

PATTERN: BARBER

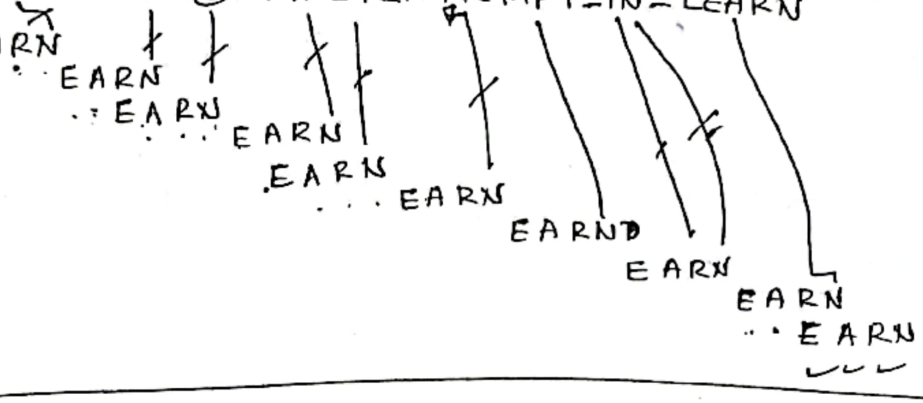


E - 1
B - 2
R - 3
A - 4

all other
Characters = 6.

TEXT : FAILS_MEANS_FIRST_ATTEMPT_IN_LEARN

PATTER: EARN



Shift table

R - 1
A - 2
E - 3
Other - 4

Algorithm ShiftTable(P[0..m-1])

// Fills the Shift table
// input : pattern P[0..m-1]
Initialise all the ele of table with m.
for (j ← 0 to m-2) do Table[P[j]] ← m-1-j

return table.

Algorithm Horspool(P[0..m-1], T[0..n-1]

```
ShiftTable(P)
 i ← m-1
 while(i<=n-1) do
    K ← 0   // No of matched Characters.
    while(K <= m-1 and P[m-1-K] == T[i-K] do
      K ← K+1
    if K==m
        return i-m+1
    else
        i ← i + Table[T[i]]
 return -1
```

Boyer Moore's String Matching.

2 table.

Bad Symbol table.

Good Suffix table.

Bad Symbol ~~table~~ Shift (d_1)

Text : _ _ _ _ _ _ S E R _ _

Pattern: B A R B E R

No of Characters matching K = 2.

Next iteration No of Shift depends on S not on R

$t(S) = 6 - K = 6 - 2 = 4$

In harpsool it was R which decide Shift.

Text : _ _ _ _ _ _ A E R _ _

Pattern: B A R B E R

$d_1 = t(A) - K$

$= 4 - 2$

$= 2$

In harspool shift value of R - 3.

here it is Shifted to 2 times.

Value $t(c) - K$ must be > 0, if $t(c) - K \leq 0$,

then $d_1 = \max\{t(c) - K, 1\}$

↓ Bad-Symbol Shift.

Good suffix Shift(d_2) – is created based on pattern.

Pattern is ~~ABCAB~~ ABCBAB

K	patter	d_2
1	ABCBAB	

K	pattern	d_2	
1	ABCBAB	2	(1 char as suffix
2	ABCBAB	4	(where AB is there is prefix)
3	ABCBAB	4	
		K=3	
4	ABCBAB	4	

Ex:

K	Pattern	d_2
1	BAOBAB	2
2	BAOBAB	5
3	BAOBAB	5

(AB is not present in prefix, so check whether B is present, yes it is present, but we can't consider 1st B because before B there is 0, so, check whether another B is present in prefix, yes, what is the position of B is 5

Ex

Text: ----XBAB--
Pattern: DBCBAB

BAB or AB is not there in prefix so shift by length of pattern.

----XIII---
ABCBAB
ABCB

Part of suffix AB is there in prefix length is 4 so shift by 4.

Example

① TEXT : B E S S _ K N E W _ A B O U T

Pattern : B A O B A B

③ K = 0

$d_1 = t(k) - k$

$= 6 - 0$

$= 6$.

B A O B A B

K = 2

$d_1 = 6 - 2 = 4$

$d_2 = 5$

B A O B A B

K = 1

$d_1 = 6 - 1 = 5$

$d_2 = 2$

$\max(d_1, d_2) = 5$

B A O B A B

① t.

Shift table

A	1
B	2
O	3
D, E, F, ...	6.

② Good Suffix table.

K	Pattern	d_2
1	BAOBAB	2
2	BAOBAB	5.
3	BAOBAB	5.
4	BAOBAB	5

If char does not match after some comparison the shift size d is calculated as

$$d = \begin{cases} d_1, & \text{if } k = 0 \\ \max\{d_1, d_2\} & \text{if } k > 0. \end{cases}$$

Algorithm:

1. for the given pattern and the alphabet used in both Pattern and text, Construct a bad Symbol Shift d_1 as $d_1 = t(c) - k$; where t is Shift table & k i's the no of matching characters.

   if d_1 is negative, then shift size Should be

   1. $d_1 = \max\{ t(c) - k, 1\}$

2. Using the pattern, Construct the good Suffix Shift d_2
3. Align the pattern against beginning of the text
4. Repeat these steps until a matching Substring is found or the pattern reaches beyond the last character of text

   Starting with the last Character of the Pattern, Compare the corresponding Characters of pattern & text.

   If Characters does not match after Comparison the Shift Size d is Calculated as

$$d = \begin{cases} d_1 & \text{if } K = 0 \\ \max(d_2, d_1) & \text{if } K > 0 \end{cases}$$

EX

TEXT : STOU_TOR_STOR

Pattern : STOR

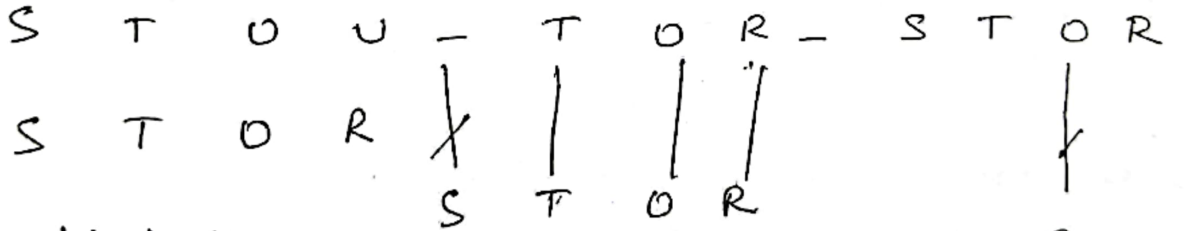
Good Suffix table.

K	pattern	d2
1	STO<u>R</u>	4
2	ST<u>OR</u>	04
3	S<u>TOR</u>	04.

Bad Shift table.

O - 1
T - 2
S - 3
other = 4

len = 5



$d_1 = t(U) - K$
$4 - 0 = 4$

$d_1 = t(\_) - K$
$= 4 - 3 = 1$

$d_2 = 4$

$d_1 = t(O) - O$
$= 1 - 0$
$= 1$

gadji_beri_bimba_glandridi

ajiji

good shift table.

K	Patter	d_2
1	ajiji	2
2	ajiji	2
3	ajiji	5
4	ajiji	5

$d_1 = t(d) - K = 5 - 2 = 3$

$d_2 = 2$ $\max(3,2) = 3$

$d_1 = t(e) - 0 = 5 - 0 = 5$

$d_1 = t(b) - 1$

$= 5 - 1$

$= 4$

$d_2 = 2$

$\max = 4$

$d_1 = t(-) - 0$

$= 5 - 0$

$= 5$

$d_1 = t(d) - 0$

$= 5 - 0$

$= 5$

No match

a j i j i

a j i j i

a j i j i

a j i j i

Text

Pattern: dridi

1	dridi	2
2	dridi	5
3	dridi	5
4	dridi	5

~~d[= tf~~