(i) Boyer-Moore Algorithm :- 1. Bad Match Table
2. Good shift Rule.

It is a pattern matching Algorithm.

Ex: Q :- T = WELCOMETOSREYASCOLLEGE

P = SREYAS

finding length

Step 1 :- $n = |n| = 22$ [length of text]

$m = |m| = 6$ [length of pattern]

Step 2 :- Construct a Bad Match Table (Only for Pattern)

S	R	E	Y	A	*
5	4	3	2	6	6

Rule 1 :- value of '*' is length of pattern

Rule 2 :- last letter of the table holds the value of * only.

$\boxed{\text{Value} = \text{length} - \text{Index} - 1}$

S ⇒ 6 − 0 − 1 = 5.

R ⇒ 6 − 1 − 1 = 4.

E ⇒ 6 − 2 − 1 = 3

Y ⇒ 6 − 3 − 1 = 2

Step 3 :- We always match the Character from Right to left

Ex:- WELCOME TO SREYAS COLLEGE
SREYAS [Compare M & S] they are mismatched

→ We see the value of M from the bad Match Table, if it is not there then we select '*'. we have to shift pattern for '*'(6) spaces towards right by leaving the first character i.e., 'W'.

WELCOMETOSREYASCOLLEGE
SREYAS [E & S are Mismatched] Find value of E in Bad Match table] we found E has the value 3 so move right 3 spaces by ignoring 1st character 'E'.

WELCOMETOSREYASCOLLEGE
SREYAS

Step 4:- The pattern P is found at the 9th position of the text (T).

Example 2:-

T = WELCOMETOSREYASCOLLEGE

P = COLLEGE

0 1 2 3 4 5 6

Sol Step 1:- finding lengths

$n = |n| = 22$

$m = |m| = 7$

Step 2:- Construct a Bad Match table

C	O	L	E	G	*
6	5	~~4~~ 3	7	1	7

If the letter is repeated, we need to calculate for all letters, but we should consider latest (or) last value

$C \Rightarrow 7 - 0 - 1 = 6$

$O \Rightarrow 7 - 1 - 1 = 5$

$L \Rightarrow 7 - 2 - 1 = 4$

$L \Rightarrow 7 - 3 - 1 = 3$

$E \Rightarrow 7 - 4 - 1 = 2$ (x)

$G \Rightarrow 7 - 5 - 1 = 1$

Note:- When there is a mismatch of any position in the Pattern we have to Consider the bad match value Only for 1st letter (R to L)

Step 3:- Performing Matching

WELCOMETOSREYASCOLLEGE
COLLEGE

[M & G is mismatched]

take the value of E from Bad Match table i.e., 7 by leaving the 1st character

WELCOMETOSREYASCOLLEGE
COLLEGE [A & E is mismatched]

A is not there in table now Consider '*' value

WELCOMETOSREYASCOLLEGE
COLLEGE [G & E is mismatched

take the value of G in Table i.e., 1 move 1 step right by leaving the 1st character

WELCOMETOSREYASCOLLEGE
COLLEGE

Step 4:- The pattern P is found at the 15th position of Text T.

Ex 1:-
T = WELCOMETOSREYASCOLLEGE
P = SREYAS

Step 1:- finding lengths
$n = |n| = 22$
$m = |m| = 6$

Step 2:- Construct a Bad Match Table

S	R	E	Y	A	*
6	4	3	2	1	6

Value = length - Index - 1

R ⇒ 6 - 1 - 1 = 4
E ⇒ 6 - 2 - 1 = 3
Y ⇒ 6 - 3 - 1 = 2
A ⇒ 6 - 4 - 1 = 1

Step 3:-
WELCOMETOSREYASCOLLEGE
SREYAS

WELCOMETOSREYASCOLLEGE
SREYAS [E is 3]

WELCOMETOSREYASCOLLEGE
SREYAS

Step 4:-
Patter P is found at 9th position of text T

Pattern P is present in the text T at 9th position

Scanned with CamScanner

(ii) Broute-force (Naive string):-

It is the Simplest method of Pattern matching.

- It performs checking all the position in the text, whether the occurence of the pattern is found or not.
- After Each Attempt the Algorithm shift the pattern by Exactly one position to right.

Example:-

1Q:- T = a c a a b c

P = a a b

Ans Step 1:- S = 0 [S - shift]

→ if match is found, check for the next char

a c a a b c

a a b

[a & a is matched]

[c & a is mismatched]

→ If mismatch, move the pattern on right side

Step 2:-
S = 1 ✓ move a shift for one position

a c a a b c

a a b

[c & a is not matched]

Step 3:-
S = 2 ✓

a c a a b c

a a b

Step 4:- The pattern P is found by shifting S = 2 in text T.

2Q T = ABCABABCDD

P = BAB

Ans Step 1:- $S = 0$

A B C A B A B C D D
B A B

[A & B is not matched]

Step 2:- $S = 1$ [shift 1 position]

A B C A B A B C D D
B A B

[C & A is not matched]

$S = 2$

A B C A B A B C D D
B A B

$S = 3$

A B C A B A B C D D
B A B

$S = 4$

A B C A B A B C D D
B A B

Step 3:- The pattern P is found in text T by shifting $s = 4$

Drawback:

ABCABABCDD

Again we are checking

3Q T = 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1

P = 0 0 0 1

Ans S = 0 ✗ 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1

S = 1 ✓ | 0 | 0 | 0 | 1 |

S = 2 ✗

S = 3 ✗

S = 4 ✗

S = 5 ✓ 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1

0 0 0 1

S = 11 ✓

(iii) KMP Algorithm:-

- It is also used for Pattern Matching.

Note :-

1. Start Index from 1

case (i) - Same as Naive String shifting 1 position

Ex:- T = a b c d e f g
P = d e f

case (ii) - T = a b c d a b c a b c d f

P = a b c d f

→ when match is found, i & j are shifted one position to right.

→ When mismatch is found, j will start from beginning. & i will start from second character.

T = a b c d a b c a b c d f

P = a b c d f

* Important thing in this Algo. is ~~(Our text is)~~ we do not happening backtracking of text.

- This Algorithm will works on proper prefix and proper Suffix.
- We have to find out / Construct π table which is also known as LPP [Longest proper prefix]

⇒ Finding proper prefixes & Proper suffixes

T = abcd

Proper prefix = { a, ab, abc }

proper Suffix = { d, cd, bcd }

→ KMP introduce a linear time Algorithm for String matching algorithm

→ It is also used for pattern matching.

Q- finding π table.

	1	2	3	4
P_1:	a	b	a	b

	1	2	3	4	5	6	7	8
P_2:	a	b	c	d	a	b	c	y

	1	2	3	4	5	6	7	8	9	10
P_3:	a	b	c	d	a	b	e	a	b	f

	1	2	3	4	5	6	7	8	9	10	11
P_4:	a	b	c	d	e	a	b	f	a	b	c

	1	2	3	4
P_1:	a	b	a	b
π	0	0	1	2

Yes in 1st position

Yes in 2nd position

	1	2	3	4	5	6	7	8
P_2:	a	b	c	d	a	b	c	y
π	0	0	0	0	1	2	3	0

	1	2	3	4	5	6	7	8	9	10
P_3:	a	b	c	d	a	b	e	a	b	f
π	0	0	0	0	1	2	0	1	2	0

	1	2	3	4	5	6	7	8	9	10	11
P_4:	a	b	c	d	e	a	b	f	a	b	c
π	0	0	0	0	0	1	2	0	1	2	3

find the pattern in the Text using KMP Algorithm.

Q. T = a b a b c a b c a b a b a b d.

p = a b a b d

Step 1 :- You have to Construct π table (only for pattern)

j	1	2	3	4	5
	a	b	a	b	d
0	0	0	1	2	0

T = m (length of Text or string)

P = n (length of pattern)

the complexity of KMP is $O(m+n)$ which is lesser than the other algo

Step 2 :- (i) Take two variables i and j where,

i = string(T) ⇒ indicates 1st location of

j = P(0)

(ii) Compare T(i) = P(j+1)

1.) If match is found (move both i & j to right side.)

2) If mis-match is found (move j to the location as per π table)

3) If j reaches to 0 (j = 0) then move i to the right side.

T(1) = P(0+1) — Proceed as per above steps.

a = a [Match is found]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	b	a	b	c	a	b	c	a	b	a	b	a	b	d

i i i

j=0	1	2	3	4	5
	a	b	a	b	d

j j j

T(i) = P(j+1) move j to b's position

T(5) = P(4+1) ⇒ c ≠ d i.e., 2

$i=5$

a b a b d

j

$T(i) \Rightarrow P(j+1)$

$T(5) = P(3)$

$c \neq a$

1	2	3	4	5
a	b	a	b	d
0	0	1	2	0

j=0 a b a b d

$j=0$ [move i to right side]

$\Rightarrow i=6$

$T(i) = P(j+1)$

$T(6) = P(0+1)$

$a = a$

$T(7) = P(1+1)$

$b = b$

$i=8 \Rightarrow T(8) = P(2+1)$

$c \neq a$

$j=0$ [move i to right side]

$i=9$

$T(9) = P(0+1)$

$a = a$

$T(10) = P(1+1)$

$b = b$

$T(11) = P(2+1)$

$a = a$

$T(12) = P(3+1)$

$b = b$

$i=13 \Rightarrow T(13) = P(\overset{j=4}{4}+1)$

$a \neq d$

$T(13) = P(a+1)$

$a = a$

$T(14) = P(3+1)$

$b = b$

$T(15) = P(4+1)$

$d = d$

$i - j$

The pattern P is found in T at position 11