String-Related Algorithms/DS

Rabin Karp

→ { KMP (Knuth-Morris-Pratt)

Z- Algorithm

Boyer- Moore

Aho- Corasick

Suffix

Tries

Given a string s.

For every index i, find the length of longest substring from it which is equal to prefix of whole string.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	x	y	z	x	x	y	z	w	x	x	y	z	x	x	y	z	x
					⇑ 1								⇑ 5				

prefix ⇒ start from 0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	x	y	z	x	x	y	z	w	x	x	y	z	x	x	y	z	x
↑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
18	1	0	0	4	1	0	0	0	8	1	0	0	5	1	0	0	1

z-array

j = 9 → 0

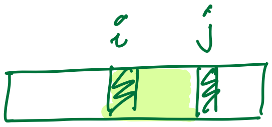
j = 10 → 1

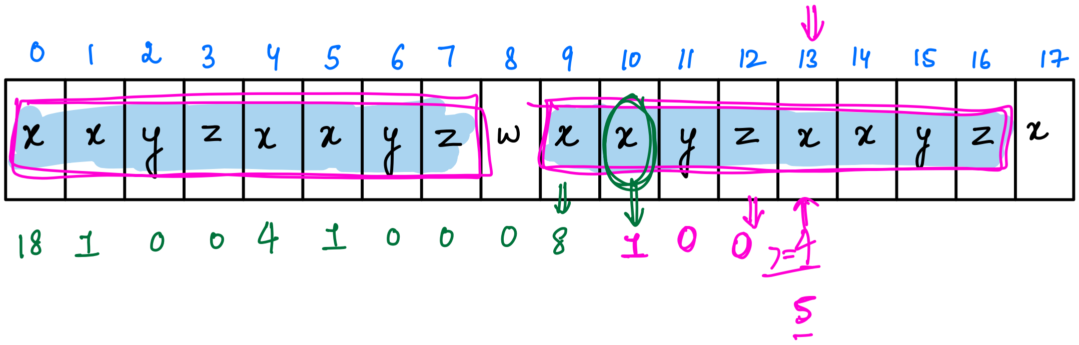
j → j - i

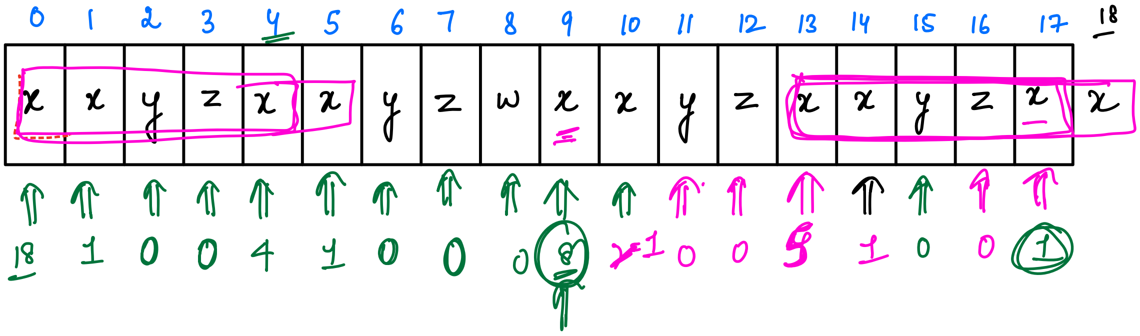
B.F:

```
z[0] = length
for( i = 1 ⟶ n)
{
    int j = i;
    while ( j < N && s[j] == s[j-i])
    {
        j++;
    }
    z[i] = j-i;
}
```

T.C : O(n^2)







0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
x x y z x x y z w x x y z x x y z x x
18 1 0 0 4 1 0 0 0 8 x 1 0 0 5 1 0 0 1

case II: inside the pink

1) no of elements within pink

copy

2) touch the boundary

3) crosses the boundary } ⬭ + BF → new pink box

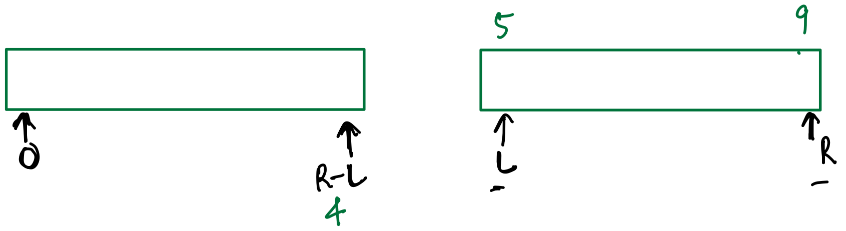
case I: when you don't know anyt

→ out of pink box

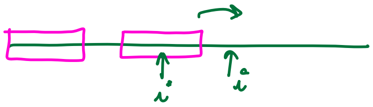
→ BF

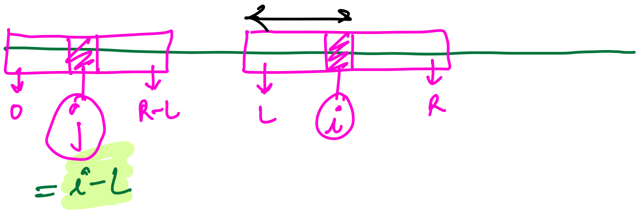
whenever you'll brute force → update the pink box

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x	x	y	x	x	y	x	x	a	x	x	y	x	x	z
⇓	⇓	⇑	⇑	⇑	⇑	⇑	⇑	⇓	⇓	⇓	⇓	⇓	⇓	⇓
	apply BF	BF ⇓	BF	copy	cc			BF ⇓	BF ⇓					
15	1	0	5	1	0	2	1	0	5	1	0	2	1	0

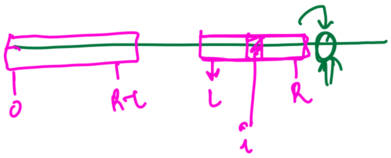


```
z[0] = length of the array
  L=0 , R=0

for( i=1 ⟶ n)
{
      // outside the segment
      if( i > R)
      {
            L=i; R=i
            while( R<N && s[R]==s[R-L])
            {
                  R++;
            }
            z[i]= R-L;
            R--;
      }
      else {
```





```
            int j = i-L;        // correspon ele

            if (z[j] < R-i+1 )
            {         z[i] = z[j]; }
            else
            {   L=i;   R++;
                while( R<N && s[R]==s[R-L]){R++; }
                 z[i] = R-L;  R--;
            }
}
```



T.C: O(n)

i: 1 ⟶ N : O(N)

L: 1 ⟶ N : O(N)

R: 1 ⟶ N : O(N)

3N = O(3N)

Q Given a string s and a pattern t. Find if there exists a substring in s which matches the pattern t.

S: my name is mohit

t: mei

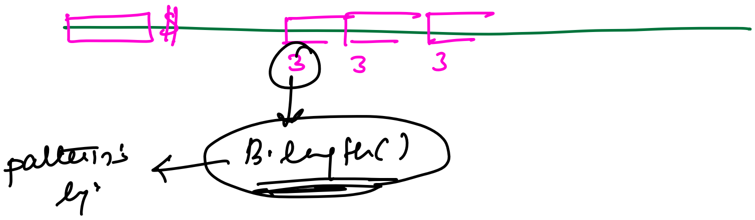
mei$ my name ismohit

m	e	i	$	m	y	n	a	m	e	i	s	m	o	h	i	t	m	e	i
17	0	0	0	1	0	0	0	3	0	0	0	1	0	0	0	0	3	0	0

T: O(n+m)

S.C: O(n+m)

Q Given two strings A & B. Find the no of times B occurs in A as a substring.



Q Given a binary string, find all the cyclic permutations of the string such that

given string

B ^ cyclic permutation = 0 .

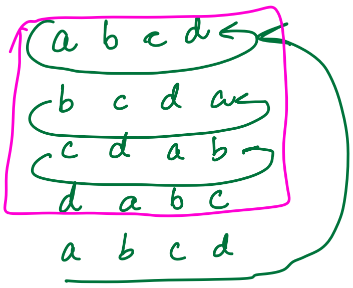
B = string

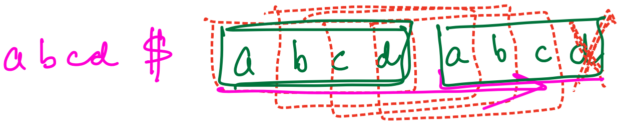
B = 1010

B = 1010
0101
1010
0101

B ^ cych = 0

ans = 2

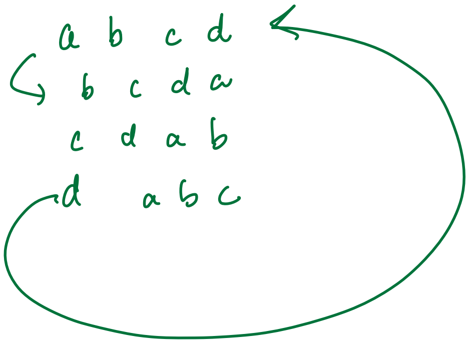




1010 $ 10101010

x0200 0 40404020

③ −1 = ②



a b c d
b c d a
c d a b
d a b c