CLASS 31

13 March 2023 21:09

CLASS-31

LPS Array

lps Array

a a a b a a

lps[i]

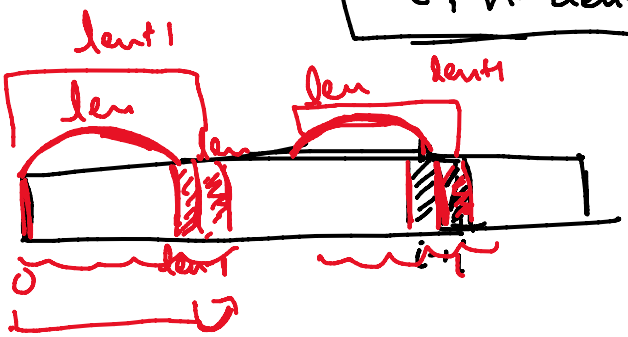
lps[n]

$O(n^3)$ (Brute force)

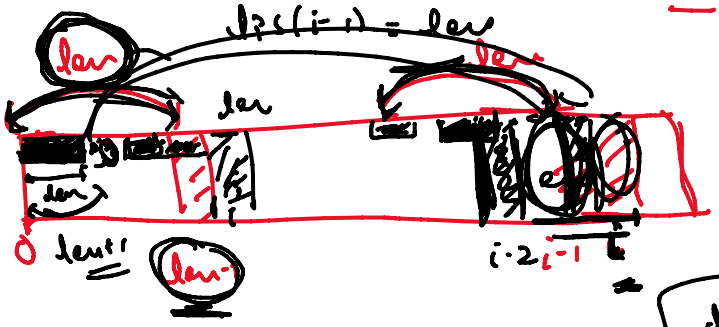
len

i-th lps[i]

i, n-lent i



lps[i] = lent+1



lps[i] = 6 x

5, 4, 3

lps[i] = len-1 ?

lps[len-1]

O(n)

```
void fillLPS (int lps[], string pat)
{
  int n = str.size();
  int len = 0;
  int i = 1;
  lps[0] = 0;
  while (i < n)
  {
    if (str[i] == str[len])
    {
      len++;
      lps[i] = len;
      i++;
    }
    else
    {
      if (len == 0)
      {
        lps[i] = 0;
        i++;
      }
      else
```

lps[len-1]

```
        }
        else
        {
            len = lps[len-1];
        }
    }
```

KMP Algorithm

Knuth Morris Pattern

Pattern Matching

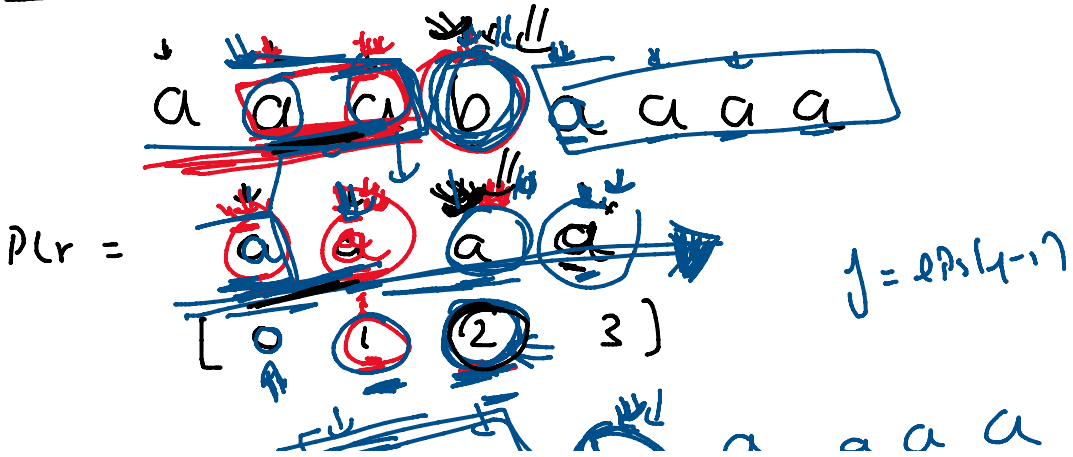
str = a a a b a c d (n)

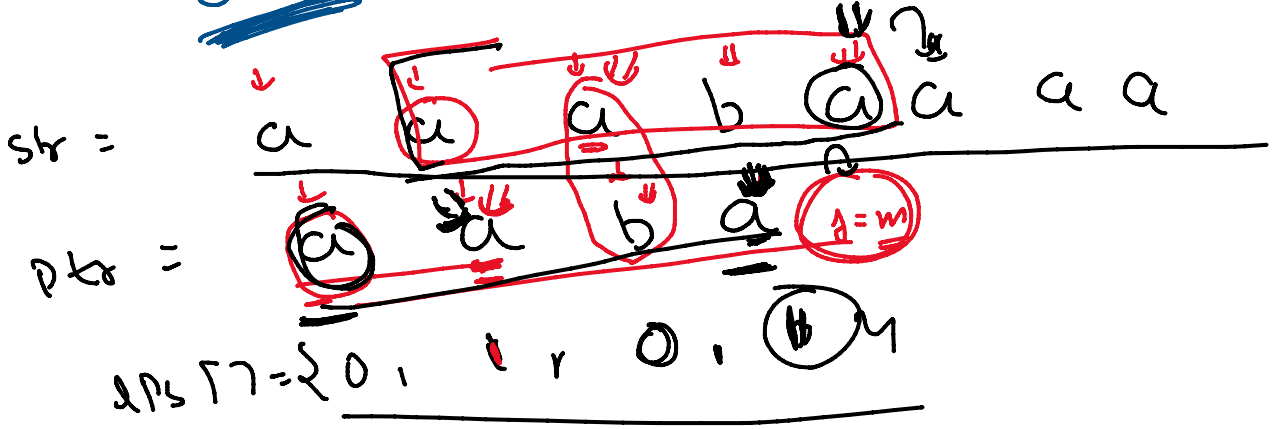
Pat = a a b (m)

Brute Force = O(n*m)

Try to optimize → KMP Algorithm (lps)

→ Rabin Karp Algorithm





lps[7] = {0, 1, 2, 0, 1, 2, 3}

```
void KMP ( String Str, String Pat )
{   int n = Str.size();
    int m = Pat.size();
    int lps[m];
    FillLPS ( lps, Pat );
    i=0 ; j=0;
    while (i<n)
    {  if ( Str[i] == Pat[j] )
       {  i++; j++;
          if(j==m)
          {  cout << " we found the Pattern" << (i-m);
             j = lps[j-1];
          }
       }
```

O(M)

T.C = O(n+m)

Brute force = O(n*m)

O(N)

$j = lps[j-1]$

}

else if (i<n and str[i] != pat[j])

{ if(j==0) i++;

else { $j = lps[j-1]$; }

}

}

Str = a a a a a a a a

Pat = a a a a =m (j==m)

lps[] = 2 0 1 1 2 3

j = 3

$j = lps[j-1]$

$j = lps[j-1] = 3$

0
1
2
3
4

Ex: Str :- A A B A A C A A B A A B A A B A

Pat → A A B A (j==m)

→ 0 1 0 1

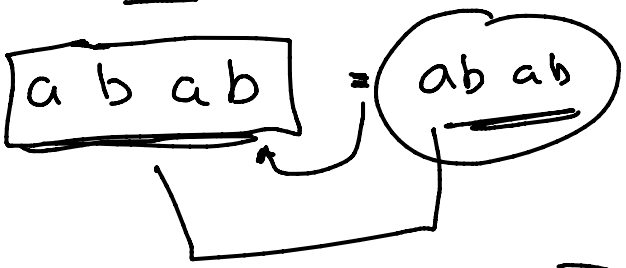
$j = lps[j-1]$

0
6
9

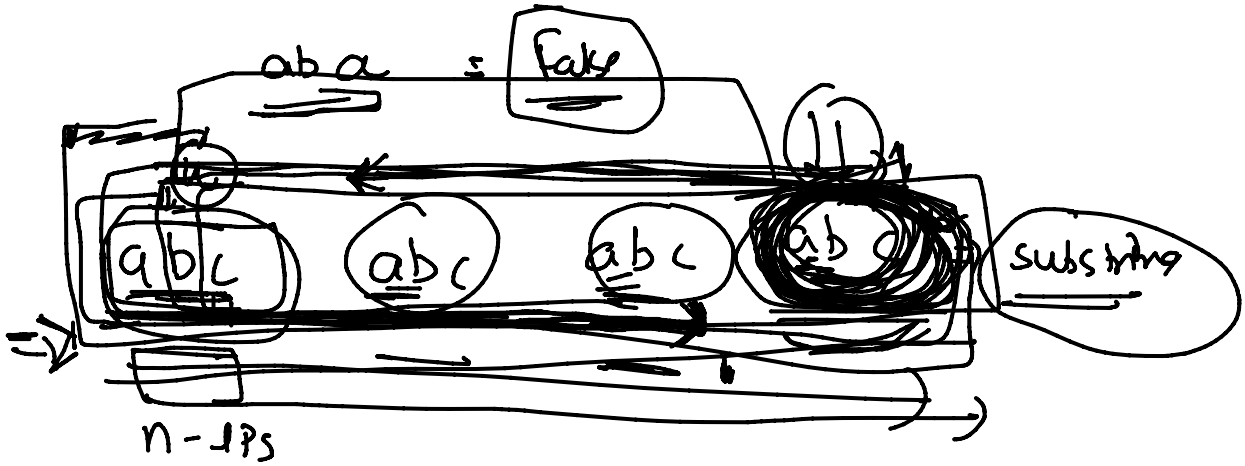
$j = lps[j-1]$

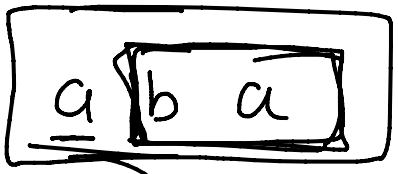
j == 0

Repeated Substring Pattern

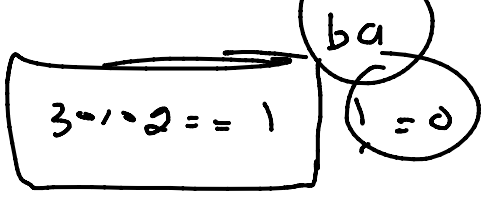


Hint KMP, LPS

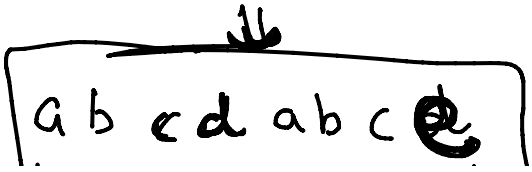


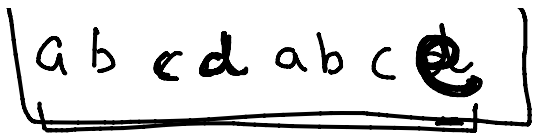


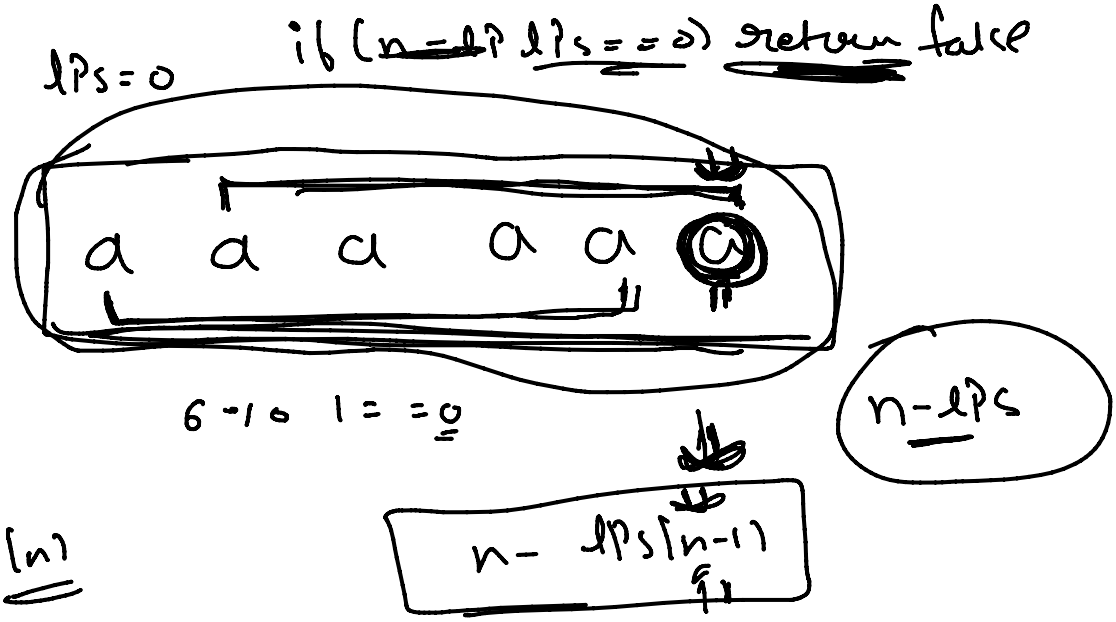
LPS = 1

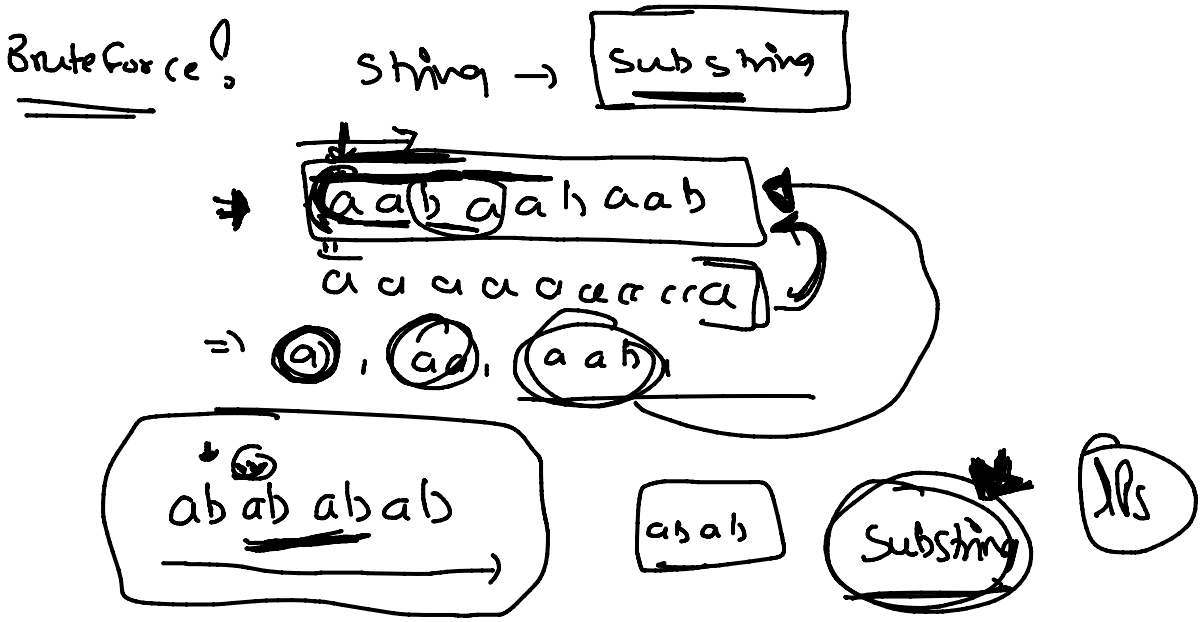


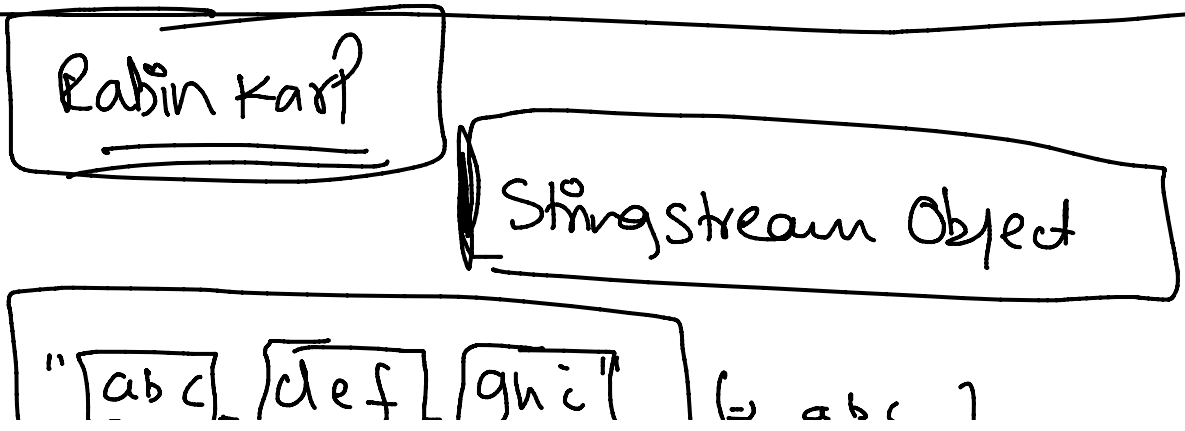
if (n % (n-LPS) == 0)




abcdabcd


if (n-lps == 0) return false
lps = 0
a a a a a a
6-1 0 1 == 0
n-lps
n- lps[n-1]
lps[n]


BruteForce!
String → Substring
aabaabaab
aaaaaaaaa
a, aa, aab
abab abab
abab
Substring
lps


Rabin Karp
Stringstream Object
"abc def ghi"
abc

"abc def ghi" => abc, def, ghi

~~For~~ (i=0; i<n; i++)

{ i=0; j=0;
while (i<n)

i, j

abc - def - ghi

{ while (str[j] != "_")
{ j++;
}
cout << str.substr(i, j-i);
i = j+1;
j++;
}