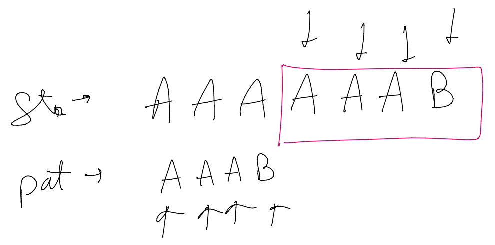
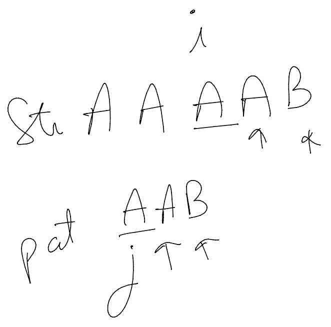
31 May 2024 21:31



1) Str, pat.

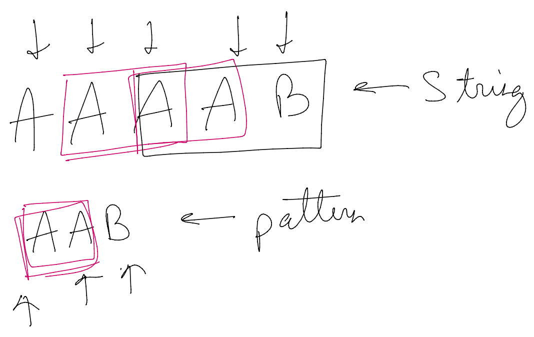
```
2) for(int i=0; i< Str.length(); i++) {
       for(int j=0; j< pat.length(); j++) {
           if( str(i+j) != pat[j] ){
               break;
           }
           if(j== pat.length()-1) {
               return true;
           }
       }
   }
   return false;
```



Time complexity: O(n×m)

n → Size of string

m → Size of pattern

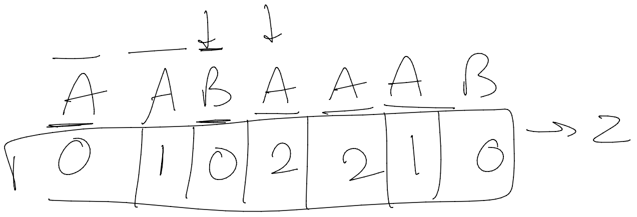


Z algo

> creation of z-array

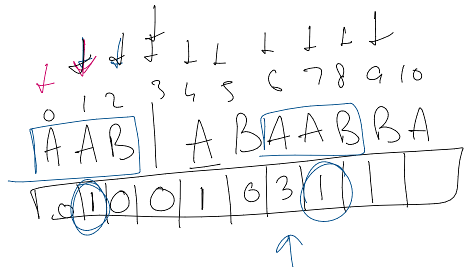
Str = A B A B A A A B B

pat = A A A B



Populating Z arr.

```
void zarr( string str, int arr[ ] ){
    int n= str.length();
    int L, R, K;
    L = R = 0;
    for (int i=1; i< str.length(); i++){
        if ( i>R ){
            L=R=i;
            while (R<n && str[R-L]== str[R]){
                R++;
            }
            arr[i]= R-L;
            R--;
        }
        else{
            K=i-L;
            if( arr [K]< R-i+1 ){
                arr[i]= arr [K];
            }
            else{
                L= i;
                while( R<n && str[R-L]== str[R]){
                    R++;
                }
                arr[ i]= R-L;
                R--;
            }
        }
    }
}
```



i	R	L
1	~~0~~ 1	~~0~~ ~~1~~
2	~~2~~ ~~2~~ ~~1~~ ~~1~~	~~2~~
3	3	3
4	~~4~~ ~~4~~ ~~3~~	~~4~~
5	5	5
6	~~6~~ ~~7~~ 8 8	~~6~~ ~~9~~
7		

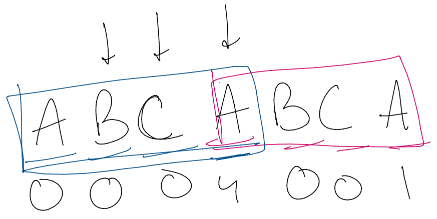
K = 7 − 6 = 1

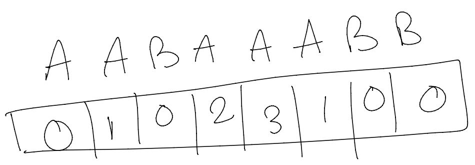
Z [i] → longest prefix starting at index i

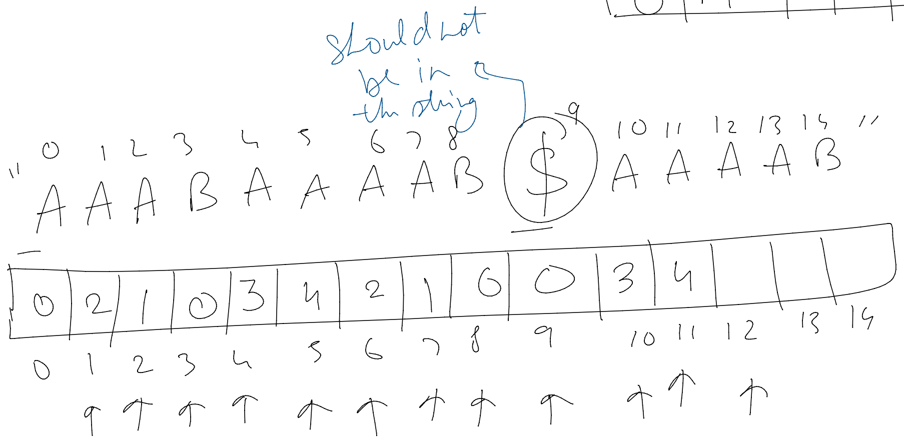
3

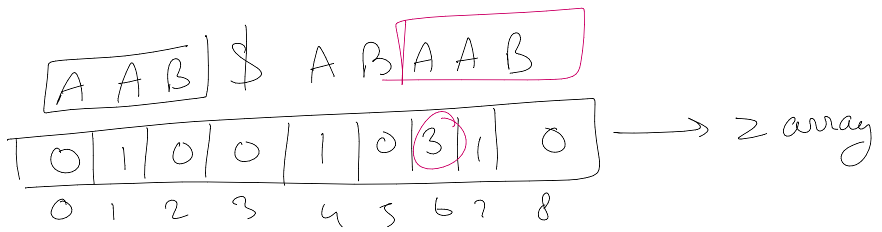
3

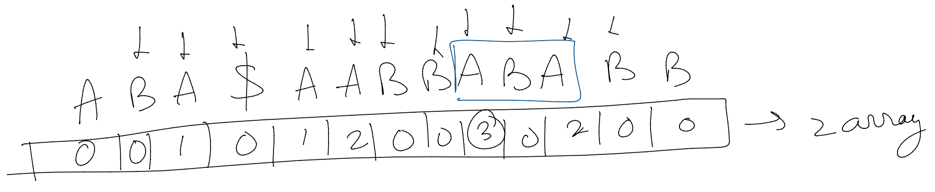
starting at index i











Sum of scores of built strings

$s = abaca$ $\quad$ $\overset{1}{s_1 = a}$ $\quad$ $\overset{0}{s_2 = ca}$

$s = abaca$ $\quad s_1 = \overset{1}{a}$ $\quad s_2 = \overset{0}{ca}$

$s_3 = \overset{1}{aca}$ $\quad s_4 = \overset{0}{baca}$

$s_5 = \overset{5}{abaca}$

$\boxed{7}$

a b a c a

0	0	1	0	1

$\boxed{5} + 1 + 1 = \boxed{7}$

a z b a z b z a z

0	0	0	3	0	0	0	2	0

$5 + 9 = \boxed{14}$

1) Create the z array.

2) Return sum of z array + s.length().