

My attempt at Z algorithm.

$S = \text{"ababadababad"} \rightarrow$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
S	a	b	a	b	a	d	a	b	a	b	a	b	a	d
Z	0	0	3	0	1	0	5	0						

So we handled the case I where

$z[i]$'s value should be actually less than $z[i-L]$'s value.

By the logic of the previous method if we try to place $z[i] = z[i-L]$

And the condition

$$(i + z[i-L] - 1 \leq R)$$

is also getting satisfied, that means that it is not escaping the window.

But These two prefixes match perfectly and if you place

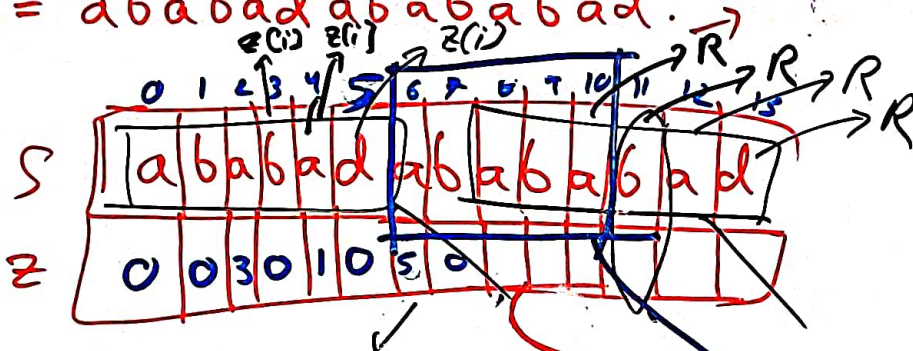
$$z[6] = z[6-6] = z[0]$$

$$= 3 \{aba\}$$

whereas ababad is matching which is completely wrong.

My attempt at Z algo rithm.

$S = \text{"ababadababad"}$



So have to think of two cases.

Case 1 (Less characters match)
 $i + Z[i-L] - 1 \leq R$
 So that it does not exceed the window

Case 2
 populating this index
 (more characters match in reality).

So this means I have to keep matching further, it means.

while am here this is in active window.

$L=6$
 $R=10$

So I can do

$Z[i] = \min(Z[i-L], R-i+1)$
 then I have to further compare.

while ($S[R+1] == S[i+Z[i]]$) {
 $R++$;
 $Z[i]++$;
}

My attempt at Z algo rithm.

$S = \text{"ababadababad"}$

0	1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	a	b	a	d	a	b	a	b	a	b	a	b
0	0	3	0	1	0	5	0	3	0	1	0	5	0

$L \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$
 $z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i] \quad z[i]$

while ($S[z[i]] == S[R+1]$) {

$z[i]++;$
 $R++;$

if ($i \leq R$) {

$z[i] = z[i-L];$
 $z[9] = z[9-6]$
 $z[9] = z[3]$

So at this point upon further

matching if $z[10] = z[10-L] = z[10-6]$

$z[i]$ increases then

$L = i$

consider it as a new window.

$= z[4]$
But that is clearly wrong, I have to figure out how L is changing.