S = "aabxaabx"

z[0] = 0

z[i] =

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| S → | a | a | b | x | a | a | b | x |
| Z → | 0 | | | | | | | |

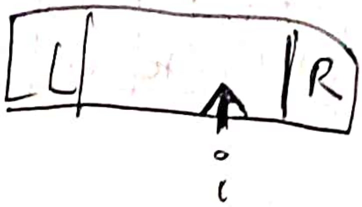
beg=0

z[0]=0

## Naive brute force

```
for (j = 1; j<N; j++){

        int k=j
        int beg=0
                     k<N && beg<N
        while (S[k] == S[beg]){

                k++;
                beg++;

        }
        Z[j] = beg

}.
```

window of [L, R] is used,




lets try to dry run ourselves.

| | 0 (L,R) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| string S → | a | a | b | x | a | a | b | x |
| Z → | 0 | 1 | | | | | | |

L=0 ; R=0

$Z[0]=0$

$\boxed{i>R}$

How to update L & R

```
for(i=1; i<N; i++){

    if(i > R){

        int k=i;
        int begin=0;

        while(k<N && begin<N && s[k]==s[begin])
        {
            begin++;
        }

        s[i]=begin;
    }
    else{

        → lets try to figure out this
          case

    }
```

L = i
R = i+begin-1


$\boxed{R-L+1}$ → window length,

s[R-L] → (this is where the prefix will be ending)

diff = R - i

$z[i] = z[R-L-(R-i)]$

$= z[R-L-R+i]$

$z[i] = z[i-L]$

Code:

```
L=0, R=0
Z[0]=0

for(i=1; i<N; i++){

    if(i>R){

        int k=i;
        int begin=0;

        while(k<N && begin<N &&
              s[k]==s[begin]){

            begin++
            k++

        }

        z[i]=begin;
        L=i;
        R=i+begin-1;
        if(R<L){R=L}.
    }
```

```
else{
    // its inside the window reese.
    Calculated solution

    z[i] = z[i-L]

}
```

{Dry running my code in the output.}

(This is the value)

{Did a mistake here $s[4] = 4$

[aabx, aabx]

$i = 6$, $L = 4$, $R = 6$

$z[i] = z[6-4]$

$= z[2]$.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| s → | a | a | b | x | a | a | b | x |
| z[] | 0 | 1 | 0 | 0 | 3 | 1 | 0 | |

$i > R$ works fine in this case

$L=0, R=0.$ (begin)

$L=1, R=1$

$i = 1$ $(i > R)\{$

$k = i$

$begin = 0$

while($s[k] == s[begin]$){

k++;

begin++;

}

$z[i] = begin$

$L = 1$

$R = 1$

}.

$i = 2$

$i > R$

$k = i$

$begin = 0$

No match

$z[i] = 0$

$L = 2$

$R = 1$

$i = 3$

$i > R$

$L = 3$

$R = 2$

$i = 4$

$i > R$

$k = 4$

$begin = 0$

$s[k] == s[begin]$

$z[i] = 3.$

$L = 4$

$R = L + z[i] - 1$

$= 6.$

$L = 4, R = 6$

$i = 5$

$i \le R$

$z[i] = z[i - L]$

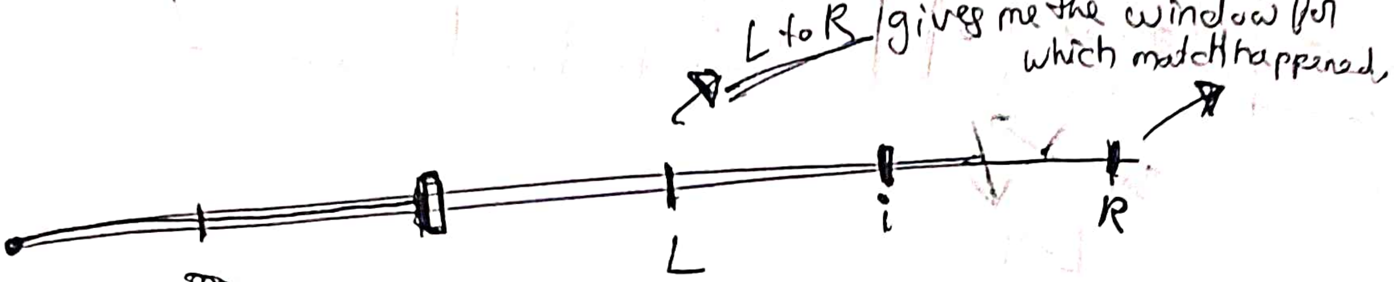
$z[5] = z[5-4].$

$z[5] = 1.$

a a a a a a a - - - - - x . . . . . . . . a a a a a a d ..

Case where it will ~~fail~~ fail,

chatgpt and the Bharat Khanna video, pointed this out to me.




$Z[(i-L)]$ has more characters matching

For example

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 (L) | 8 | 9 (R) | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | a | a | a | a | a | a | d | a | a | a | d |
| Z | 0 | 5 | 4 | 3 | 2 | 1 | 0 | 3 | | | |

$Z[i] = Z[i-L]$

$Z[8] = Z[5]$

$= 5.$

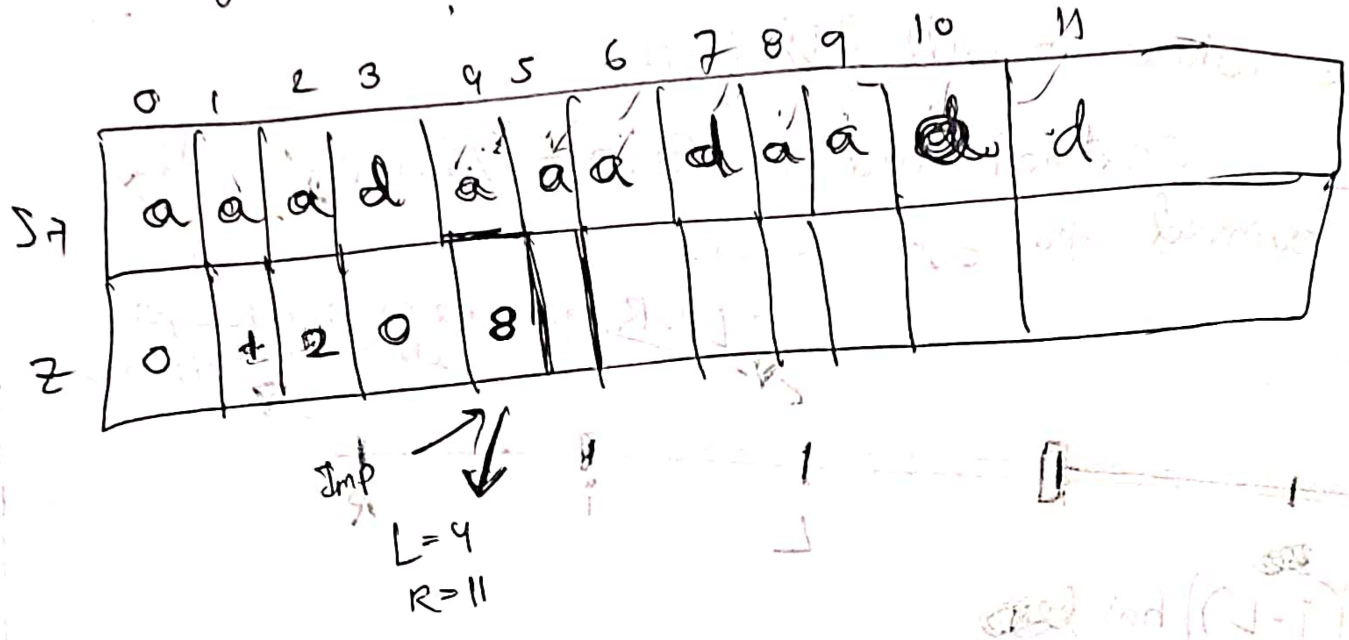
⇓

But thats wrong

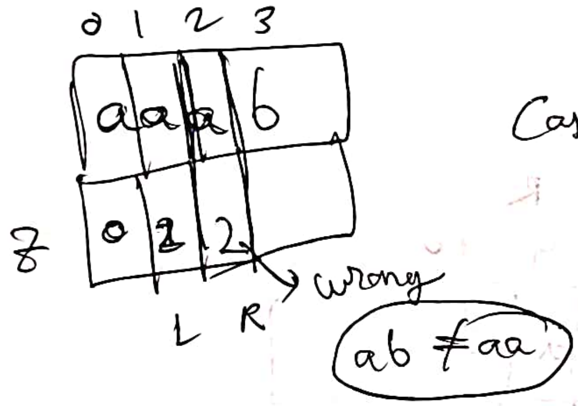
Case 1

where it does not work.

Similarly the reverse case



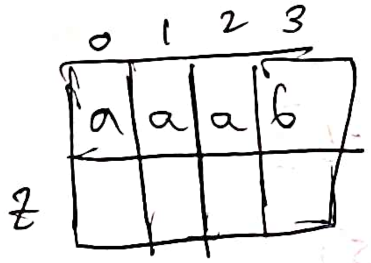

Lol, no this case works seems like I messed up.

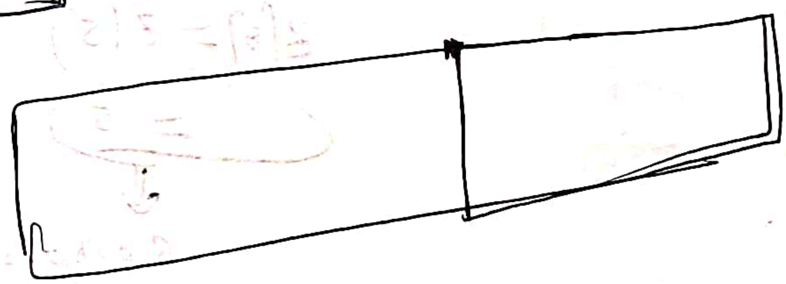



Case I where it fails

actually less characters match.

Case II more characters match



(7–9) Am

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | a | b | a | d | a | b | a | b | a | b | a | b | a | b | a | d |
| 0 | | | | | | | | | | | | | | | | | |

L R

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | a | b | a | d | a | b | a | b | a | b | a | b | a | d |
| 0 | 0 | 3 | 0 | 1 | 0 | 5 | 0 | 3 | | | | | | | |

$i < R$

$z[i] = z[i-L]$

$z[i] = z[i-L]$

$z[4] = z[2]$

$= 3$

But it's wrong

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

}

else {

$R - i + 1$

Case 1 → restrict $z[i] = R - i + 1$

Case 2

$z[i] \leq R - i + 1$

$z[i] + i$

}

L 3 R

i

$z[i]$

$i + z[i] - 1 \leq R$