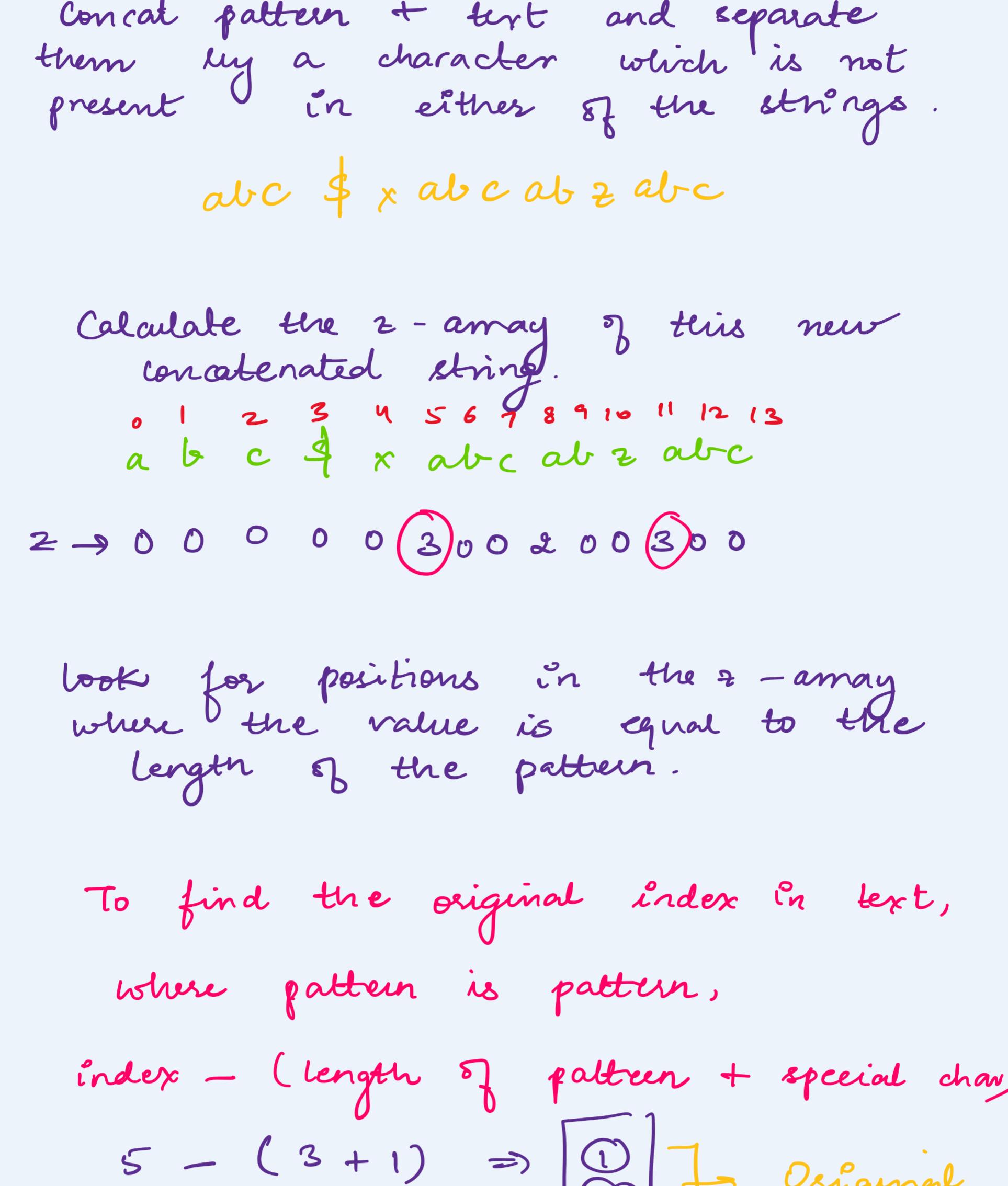


Z - Algorithm

- Naive $\rightarrow O(P \times T)$
- Rabin-Karp \rightarrow worst case $\rightarrow O(P \times T)$
Avg case $\rightarrow O(T)$

→ Core of the algorithm is the function
 $Z(k) = \text{longest substring starting at } k,$
 which is also a prefix of the string.



Q1: To use this information for pattern matching.

Q2: How to compute this Z - array effectively.

$$Q = \text{text} \rightarrow x \underset{0}{a} \underset{1}{b} \underset{2}{c} \underset{3}{a} \underset{4}{b} \underset{5}{a} \underset{6}{b} \underset{7}{c} \underset{8}{a} \underset{9}{b}$$

$$\text{pat} \rightarrow abc$$

1. Concat pattern + text and separate them by a character which is not present in either of the strings.

$$abc \$ x a b c a b z a b c$$

2. Calculate the Z - array of this new concatenated string.

$$\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ a & b & c & \$ & x & a & b & c & a & b \\ \text{Z} \rightarrow 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

3. looks for positions in the Z - array where the value is equal to the length of the pattern.

To find the original index in text, where pattern is pattern,

$$\text{index} = (\text{length of pattern} + \text{special char})$$

$$5 - (3 + 1) \Rightarrow \boxed{\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}} \rightarrow \begin{matrix} \text{Original} \\ \text{array} \\ \text{indices} \end{matrix}$$

$$\text{Time Complexity: } O(|P| + |T|)$$

length of pattern

length of text

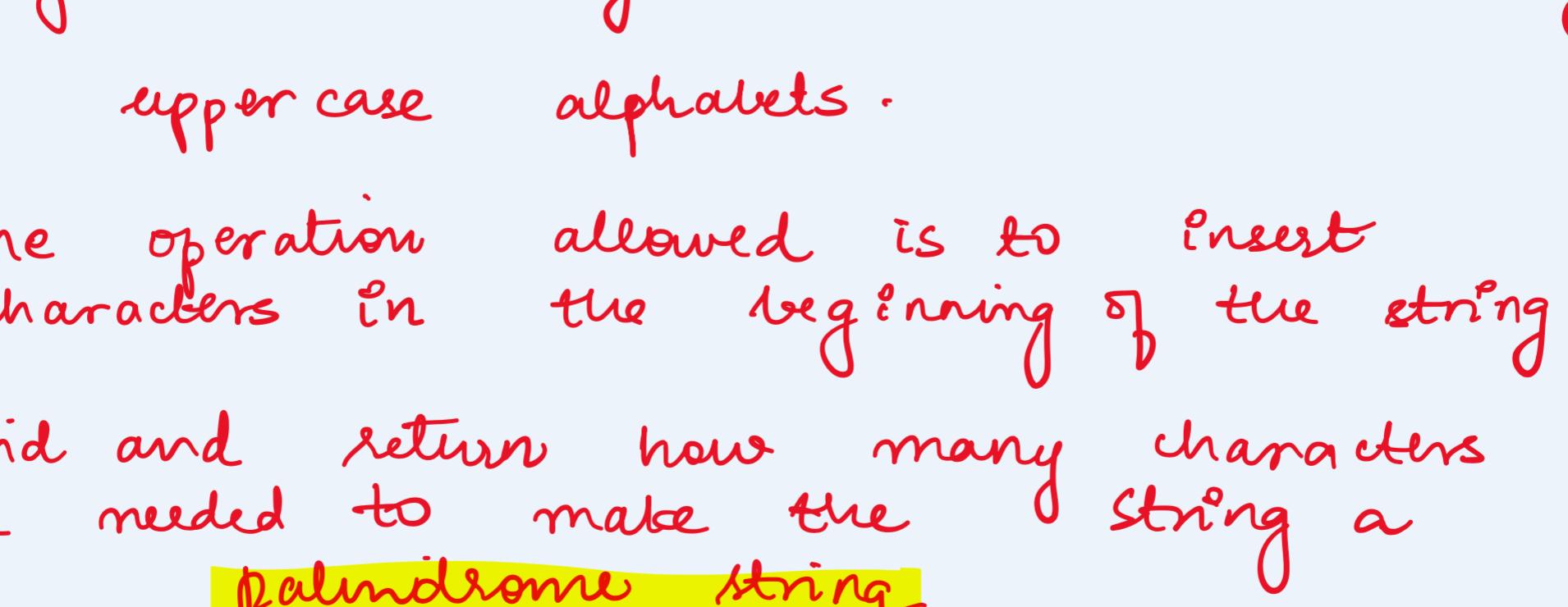
Quick ques :

$$S = P \neq T \quad \text{(No)}$$

$$\Rightarrow S = T \neq P$$

This is not a symmetric property.

Now Z - Algorithm,



Now for a given position i , where $Z_i > 0$

→ starting at position i , there is some substring (i) , that matches the prefix of the string.

$$y \neq x$$

string of length 1 is 1

$$i=0 \rightarrow i=|S|$$

Compute the $|Z|$ from $i=1$ to $i=|S|$

Naive approach

① Compute Z_1

$$\begin{array}{c} a a a a a b \\ \hline 0 3 \end{array}$$

↑?

$$\begin{array}{c} a \rightarrow a \\ a \rightarrow a \\ a \rightarrow a \\ \hline a \rightarrow b \ x x \end{array}$$

string length = 5

for just computing $Z_1 \rightarrow 9$ comparisons.

worst case

$$\begin{array}{c} Z_1 \rightarrow 4 \\ Z_2 \rightarrow 3 \\ Z_3 \rightarrow 2 \\ \vdots \end{array} \rightarrow \text{comparisons}$$

when comparing the Z - values, if the sum of Z - value + current index \geq the boundary of Z - box, then we need to recalculate the Z - value

Approach 1: \boxed{abc}

Keep removing the last char until you

find a string which is palindrome.

empty

$$\boxed{baab} \boxed{a}$$

$$\text{ans} = \boxed{1}$$

Approach 2 :

string + reverse string

$$\text{String} \rightarrow a a c e c a a a$$

$$a a c e c a a a + a a a c e c a a a$$

$$\begin{array}{ccccccccc} 0 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 0 & 1 & 2 & 0 \end{array}$$

→ find the max value of Z - array

$$\text{str length} - \text{max val} = 9 - 7 = \boxed{2}$$