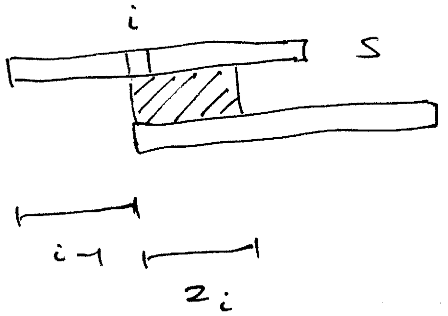
The prefix table

Given a string s, let for $i = 1, 2, \ldots, |s|$

z_i be the length of the longest common prefix of s and $s[i..|s|]$.

Figure:



(Note: $z_1 = |s|$ always.)

Example: s = ababacaaa

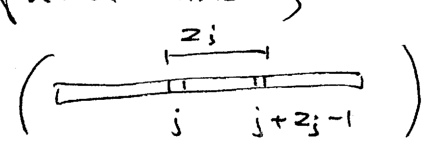
i	1	2	3	4	5	6	7	8	9
z_i	9	0	3	0	1	0	1	1	1

Below is an algorithm for finding $z_1, z_2, \ldots, z_{|s|}$.

The algorithm maintains the following invariant:

1) $z_2, z_3, \ldots z_{k-1}$ have been computed correctly

2) $r = \max\limits_{1 \le j < k} \{ j + z_j - 1 \}$



3) l = j giving the max in 2) (Some j (any) in case of multiple max's)

During the alg., we define z_1 to be 0 (not $|S|$), and set $z_1 = |S|$ in the end [this is just to ease the formulation of the invariant].

Alg. PrefixTable

$k = 2$
$r = 0$
$l = 1$
} Initialization. Fulfills inv.

while $k < |S|$:

if $k \geq r$:

Scan $S[k..]$ and $S[1..]$ simultaneously, until mismatch is found (or end of S reached).

Let t be number of matches found.

$z_k = t$
$r = k + z_k - 1$
$l = k$
} z_k has been found directly. Inv. fulfilled.

else: ($k \leq r$)

See figure *) below

$k' = k - l + 1$
$b = r - k + 1$
if $z_{k'} < b$:

$z_k = z_{k'}$
} Keep r and l. Fulfills inv.

else: ($z_{k'} \geq b$)

⋮

[else : $(z_k' \geq b)$]

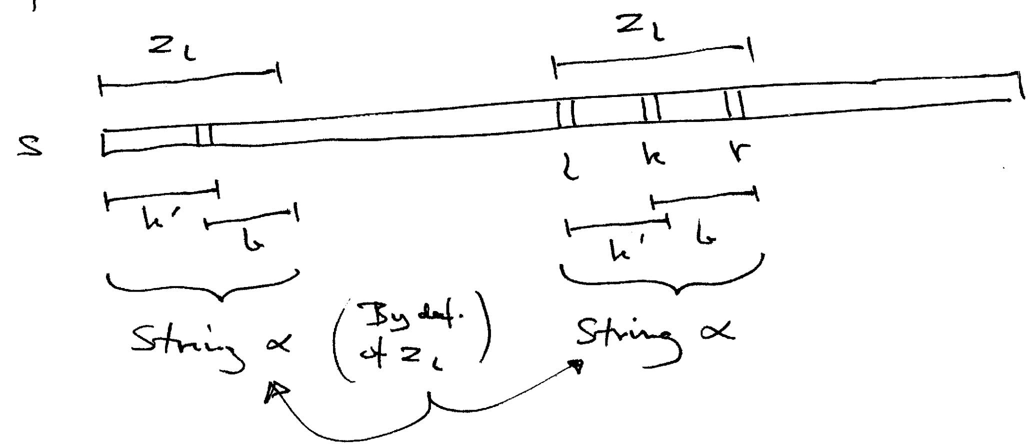
Scan $s[r+1..]$ and $s[b+1..]$ simultaneous until mismatch (or end of s reached).
Let t be number of matches found.

$$\left.\begin{aligned} z_k &= b+t \\ r &= r+t \\ l &= k \end{aligned}\right\} \text{ Inv. fulfilled.}$$

$k = k+1$ (next iteration of while loop)

$z_1 = |S|$ (finalization).

Figure *) :



Analysis : Correctness follows by maintenance of inv.
Since each iteration takes time $O(1 + \Delta r)$, and $r \leq |S|$, we have total time $O(|S|)$.