Pseudocode for Fundamental Preprocessing

The Z Algorithm[1]

Z-ALGORITHM($S[1..n]$)

```
1     ▷ INPUT: String S = S[1..n]
2     ▷ OUTPUT: (Z_k, l_k, r_k), for 2 ≤ k ≤ n

3     ▷ Base case: k = 2
4     Z_2 = n − 1                ▷ Value if no mismatch found
5     for i ← 2 to n
6         do if S(i) ≠ S(1)
7               then ▷ Have found a mismatch — a character not matching S(1)
8                    Z_2 ← i − 2
9                    break
10    if Z_2 > 0
11      then l_2 ← 2
12           r_2 ← Z_2 + 1
13      else l_2 ← r_2 ← 0

14    ▷ General case: 3 ≤ k ≤ n
15    for k ← 3 to n
16        do if r_{k−1} < k                  ▷ Have not matched S(k) yet
17              then Z_k = n − k + 1         ▷ Value if no mismatch found
18                   for i ← k to n
19                       do if S(i) ≠ S(i − k + 1)
20                             then          ▷ Have found a mismatch
21                                  Z_k ← i − k
22                                  break
23                   if Z_k > 0
24                     then l_k ← k
25                          r_k ← k + Z_k − 1
26                     else l_k ← r_k ← 0
```

CONTINUED

[1] Gusfield, Section 1.4.

Z Algorithm Concluded

```
27          else                                ▷ k is in the Z-box [l_{k-1}..r_{k-1}]
28              k' ← k − l_{k-1} + 1
29              α ← S[l_{k-1}..r_{k-1}]         ▷ S[1..r_{k-1} − l_{k-1} + 1] = α
30              β ← S[k..r_{k-1}]               ▷ S[k'..r_{k-1} − l_{k-1} + 1] = β
31              if Z_{k'} < |β|
32                then Z_k ← Z_{k'}
33                     l_k ← l_{k-1}
34                     r_k ← r_{k-1}
35                else                          ▷ Z_{k'} ≥ |β|
36                     if r_k − 1 = n
37                       then q ← n
38                       else for q ← r_{k-1} + 1 to n
39                                do if S(|β| + q − r_{k-1}) ≠ S(q)
40                                     then q ← q − 1
41                                          break
42                     Z_k ← q − k + 1
43                     l_k ← k
44                     r_k ← q
45      return (Z_k, l_k, r_k), for 2 ≤ k ≤ n
```