Algorithm 1: The Algorithm Based on Corollary 1.

---

1 **Function** CheckByFP(x, sa, lcp, n)
2 $ST_1 := [(sa[i], i, null)|i \in [0, n)]$
3 $ST_2 := [(sa[i] + lcp[i + 1], i, null, null)|i \in [0, n - 1)]$
4 $ST_3 := [(sa[i] + lcp[i], i, null, null)|i \in [1, n)]$
5   sort tuples in ST_1, ST_2 and ST_3 by 1st component
6 $fp := 0$
7   **for** $i \in [0, n]$ **do**
8   **if** ST_1.notEmpty() and ST_1.top().$1st = i$ **then**
9 $e := ST_1$.top(), ST_1.pop(), $e.3rd := fp$, ST_1'.push(e)
10   **end**
11   **else**
12   **return** false   // condition (1) is violated
13   **end**
14   **while** ST_2.notEmpty() and ST_2.top().$1st = i$ **do**
15 $e := ST_2$.top(), ST_2.pop(), $e.3rd := fp$, $e.4th := x[i]$, ST_2'.push(e)
16   **end**
17   **while** ST_3.notEmpty() and ST_3.top().$1st = i$ **do**
18 $e := ST_3$.top(), ST_3.pop(), $e.3rd := fp$, $e.4th := x[i]$, ST_3'.push(e)
19   **end**
20 $fp := fp \cdot \delta + x[i] \mod P$ // $x[n]$ is the virtual character
21   **end**
22   sort tuples in ST_1', ST_2' and ST_3' by 2nd component.
23   **for** $i \in [1, n)$ **do**
24 $fp_1 := ST_1'$.top().$3rd$, ST_1'.pop(), $fp_2 := ST_2'$.top().$3rd$, $ch_1 := ST_2'$.top().$4th$, ST_2'.pop()
25 $\hat{fp_1} = fp_2 - fp_1 \cdot \delta^{lcp[i]} \mod P$
26 $fp_1 := ST_1'$.top().$3rd$, $fp_3 := ST_3'$.top().$3rd$, $ch_2 := ST_3'$.top().$4th$, ST_3'.pop()
27 $\hat{fp_2} = fp_3 - fp_1 \cdot \delta^{lcp[i]} \mod P$
28   **if** $\hat{fp_1} \neq \hat{fp_2}$ or $ch_1 \geq ch_2$ **then**
29   **return** false   // condition (2) or (3) is violated
30   **end**
31   **end**
32   **return** true