```
COMPUTE-LCP(T, SA, n)
 1  allocate arrays rank[1 : n] and LCP[1 : n]
 2  for i = 1 to n
 3       rank[SA[i]] = i                    // by definition
 4  LCP[1] = 0                              // also by definition
 5  l = 0                                   // initialize length of LCP
 6  for i = 1 to n
 7       if rank[i] > 1
 8            j = SA[rank[i] − 1]           // T[j :] precedes T[i :] lexicographically
 9            m = max{i, j}
10            while m + l ≤ n and T[i + l] == T[j + l]
11                 l = l + 1                // next character is in common prefix
12            LCP[rank[i]] = l              // length of LCP of T[j :] and T[i :]
13            if l > 0
14                 l = l − 1                // peel off first character of common prefix
15  return LCP
```