Input: table L and sorted array H of SPM-relevant suffixes β
functions *process_leafedge*, *process_branchedge* and *process_lcpinterval* to process the enumerated items

Output: Processed implicit branch- and leaf-edges of the lcp-interval tree in bottom-up order

```
1:  stack := [ ]                                                    ▷ empty stack
2:  lastitv := ⊥
3:  firstedgefromroot := true                                       ▷ No edge from root yet
4:  stack.push(0, 0, ⊥)                                             ▷ Push root-interval with undefined rb-value
5:  for e := 0 to β − 2 do
6:      if L_{e+1} ≤ stack.top.lcp then                             ▷ new
7:          if stack.top.lcp > 0 then                               ▷ new
8:              firstedge := false                                  ▷ new
9:          else                                                    ▷ new
10:             firstedge := firstedgefromroot                      ▷ new
11:             firstedgefromroot := false                          ▷ new
12:         process_leafedge(firstedge, stack.top, H_e)             ▷ new
13:     while L_{e+1} < stack.top.lcp do
14:         lastitv := stack.pop
15:         lastitv.rb := e
16:         process_lcpinterval(lastitv)
17:         if L_{e+1} ≤ stack.top.lcp then
18:             if stack.top.lcp > 0 then                           ▷ new
19:                 firstedge := false                              ▷ new
20:             else                                                ▷ new
21:                 firstedge := firstedgefromroot                  ▷ new
22:                 firstedgefromroot := false                      ▷ new
23:             process_branchedge(firstedge, stack.top, lastitv)
24:             lastitv := ⊥
25:     if L_{e+1} > stack.top.lcp then
26:         if lastitv ≠ ⊥ then
27:             stack.push(L_{e+1}, lastitv.lb, ⊥)
28:             process_branchedge(true, stack.top, lastitv)
29:             lastitv := ⊥
30:         else
31:             stack.push(L_{e+1}, e, ⊥)
32:             process_leafedge(true, stack.top, H_e)              ▷ new
```