RS\_vEB\_TREE\_INSERT(*V*, *x*)

1. **if** *V.min* == NIL
2. vEB\_EMPTY\_TREE\_INSERT(*V*, *x*)
3. **else**
4. **if** *x* < *V.min*
5. exchange *V.min* with *x*
6. **if** *V.u* > 2
7. **if** *V.summary* == NIL
8. *V.summary* = CREATE\_NEW\_RS\_vEB\_TREE(↑√V.u)
9. *insert\_cluster* = *lookup*(*V.cluster*, *high*(*x*))
10. **if** *insert\_cluster* == NIL
11. *insert\_cluster* = CREATE\_NEW\_RS\_vEB\_TREE(↓√*V.u*)
12. *V*.*cluster*.*insert*(*insert\_cluster*) // hash table insertion
13. **if** vEB\_TREE\_MINIMUM(*insert\_cluster*) == NIL
14. RS\_vEB\_TREE\_INSERT(*V.summary*, *high*(*x*))
15. vEB\_EMPTY\_TREE\_INSERT(*insert\_cluster*, *low*(*x*))
16. **else**
17. RS\_vEB\_TREE\_INSERT(*insert\_cluster*, *low*(*x*))
18. **if** *x* > *V.max*
19. *V.max* = *x*

RS\_vEB\_TREE\_SUCCESSOR(*V*, *x*)

1. **if** *V.u* == 2
2. **if** *x* == 0 && *V.max* == 1
3. **return** 1
4. **else**
5. **return** NIL
6. **elseif** *V.min* != NIL && *x* < *V.min*
7. **return** *V.min*
8. **else**
9. *max\_low* = vEB\_TREE\_MAXIMUM(*lookup*(*V.cluster*, *high*(*x*)))
10. **if** *max\_low* != NIL && *low*(*x*) < *max\_low*
11. *offset* = RS\_vEB\_TREE\_SUCCESSOR(*lookup*(*V.cluster*, *high*(*x*)), *low*(*x*))
12. **return** *index*(*high*(*x*), *offset*)
13. **else**
14. **if** *V.summary* != NIL
15. *succ\_cluster* = RS\_vEB\_TREE\_SUCCESSOR(*V.summary*, *high*(x))
16. **if** *succ\_cluster* == NIL
17. **return** NIL
18. **else**
19. *offset* = vEB\_TREE\_MINIMUM(*lookup*(*V.cluster*, *succ\_cluster*))
20. **return** *index*(*succ\_cluster*, *offset*)
21. **else**
22. **return** NIL