1a) F

1b) T

1c) T

1d) F

1e) F

1f) T

1g) T

1h) F

1i) F

1j) F

1k) F

1L) F

1m) T

1n) T

1o) F

1p) T

2) 0->1->2->3

1->5

2 -> 5

3

4

5

6

7

3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | F | F | T | T | F | F | F | F |
| 1 | F | F | T | F | F | F | F | f |
| 2 | F | F | F | f | F | F | F | F |
| 3 | F | F | F | F | F | T | F | F |
| 4 | F | F | F | F | F | F | T | F |
| 5 | F | F | F | F | F | F | F | T |
| 6 | F | F | F | F | F | F | F | T |
| 7 | F | F | F | F | F | F | F | f |

4) output: 0,1,4,2,3,6,5,7

The possibilities for the first node in the topological sort above is 3

5)

6)The bug with this code is that it never updates the value, n.

7) You can convert a BST into an AVL tree worst case by traversing the original tree in O(n) and inserting elements into the empty AVL tree