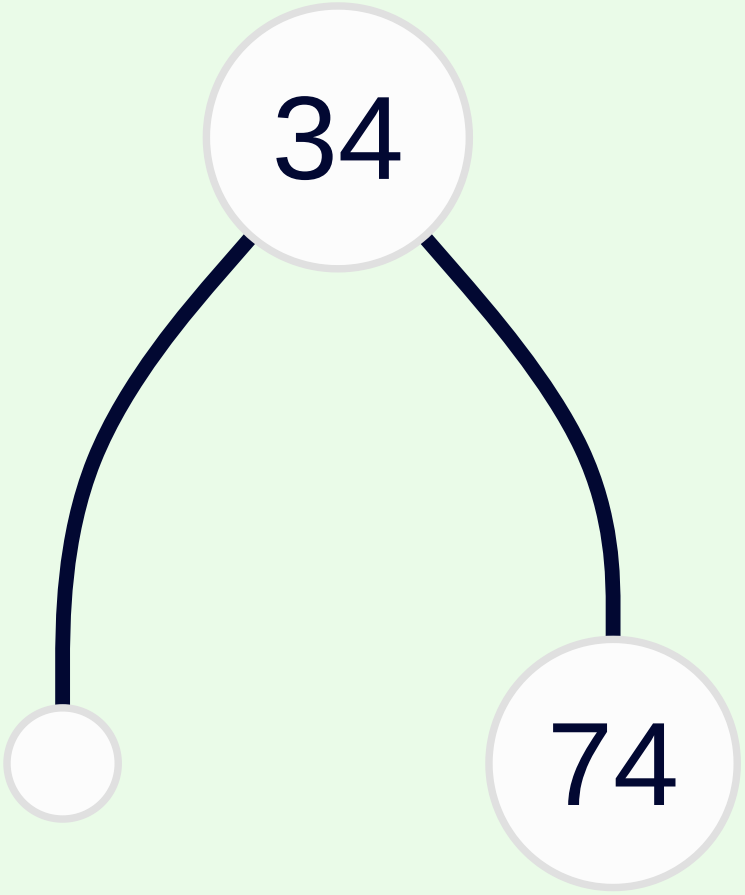
1

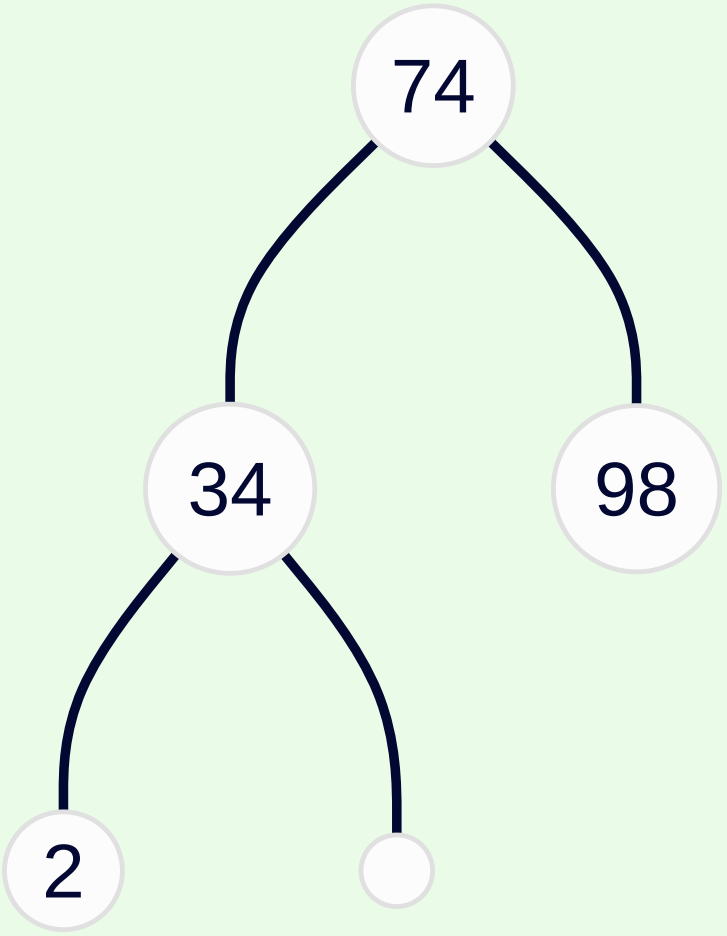
AVL Trees

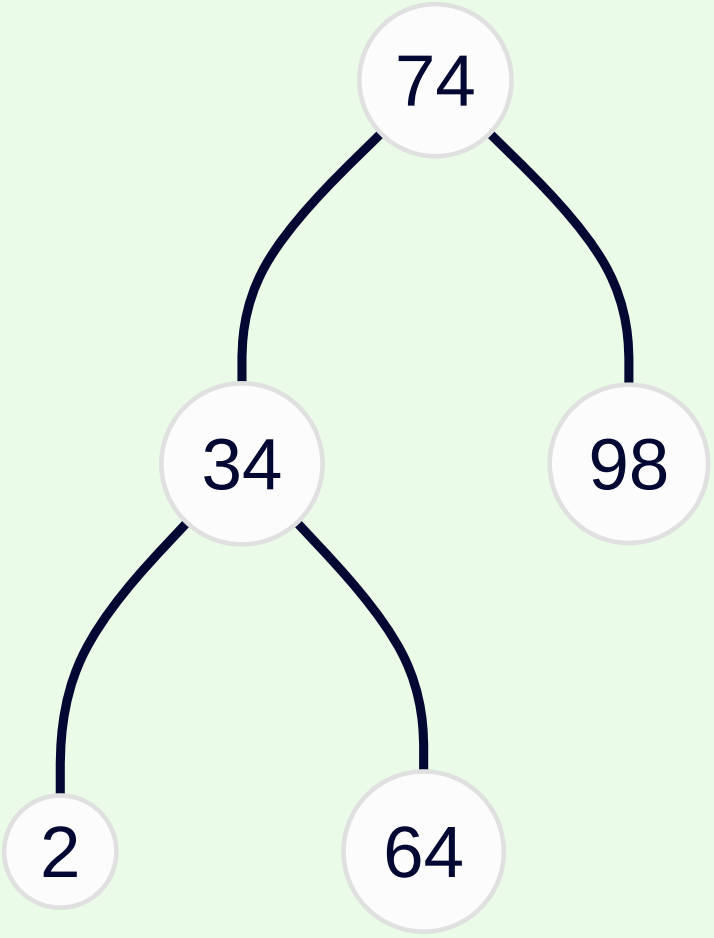
a

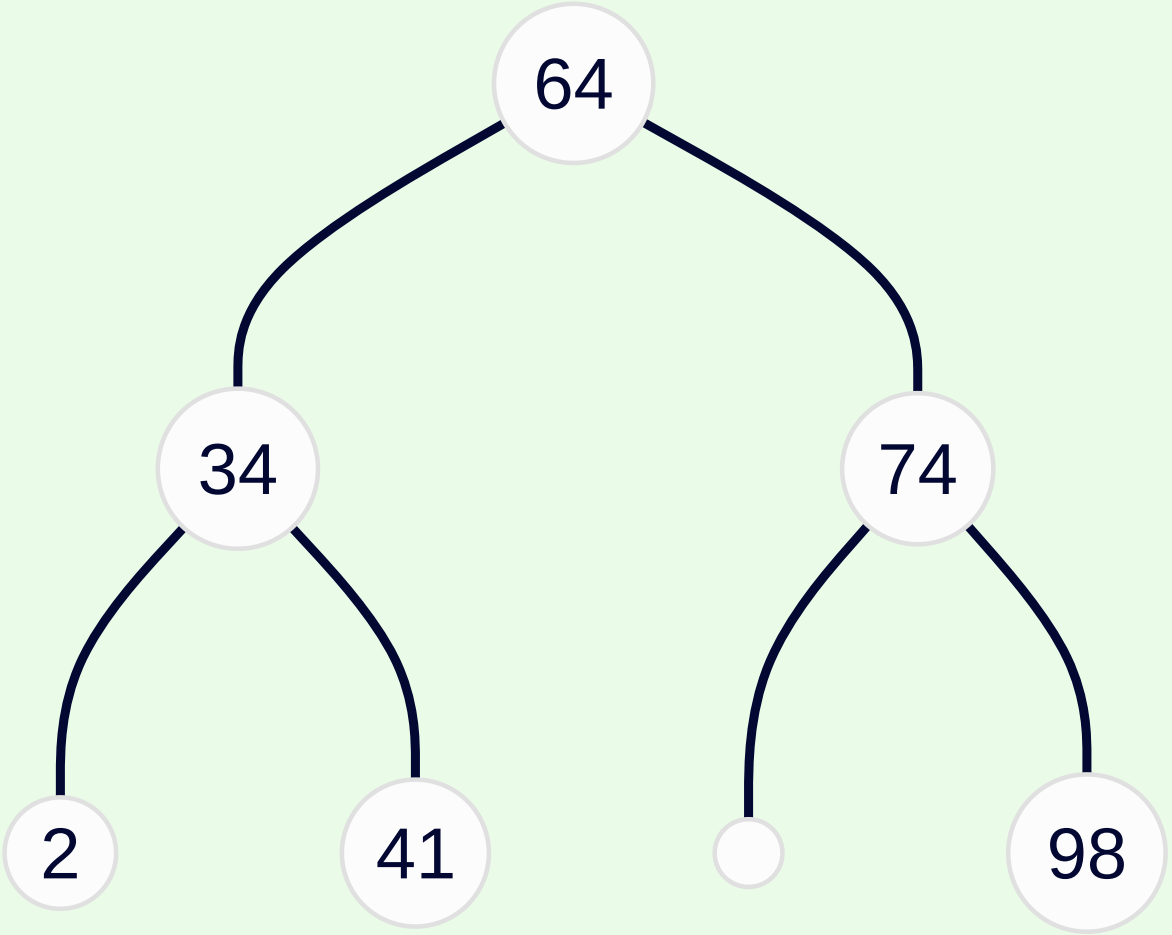
Insert the following values in the given order [34, 74, 98, 2, 64, 41, 86, 59, 40, 58, 50, 56, 45, 16, 44] into an empty AVL tree. Show the tree after all insertions.

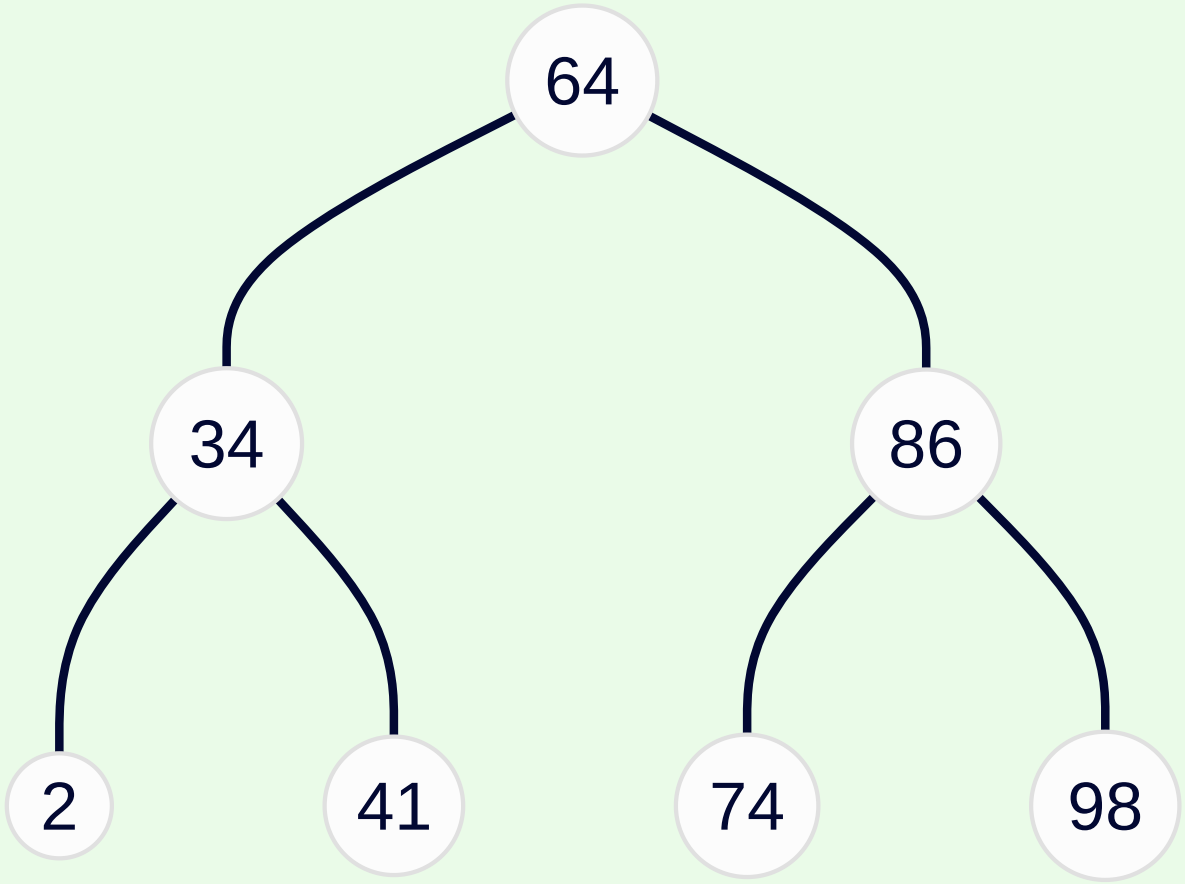
✓ Answer

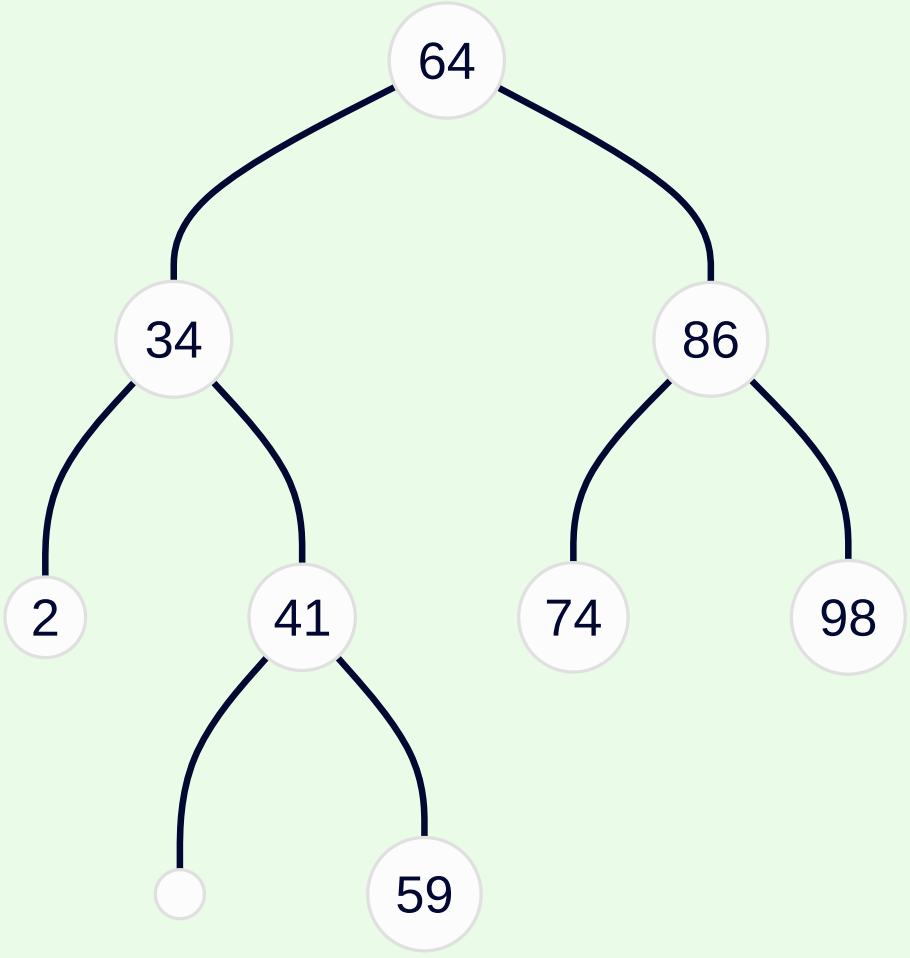

34
74

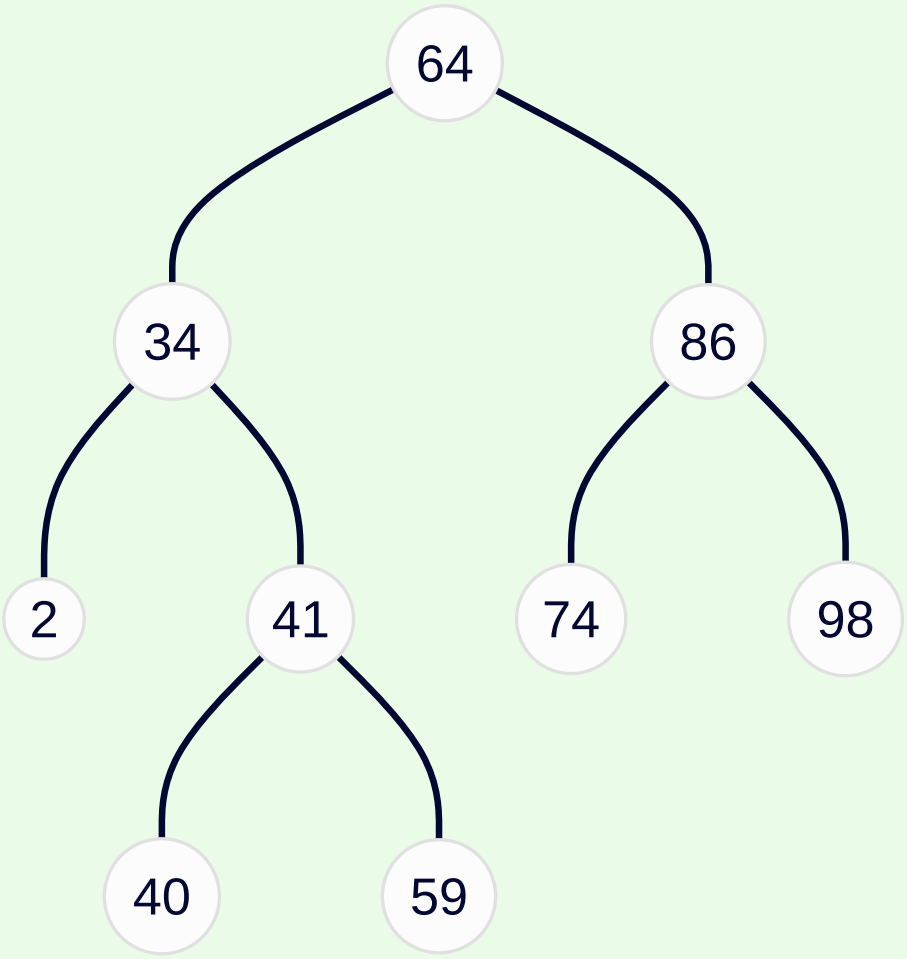

74
34
98
2

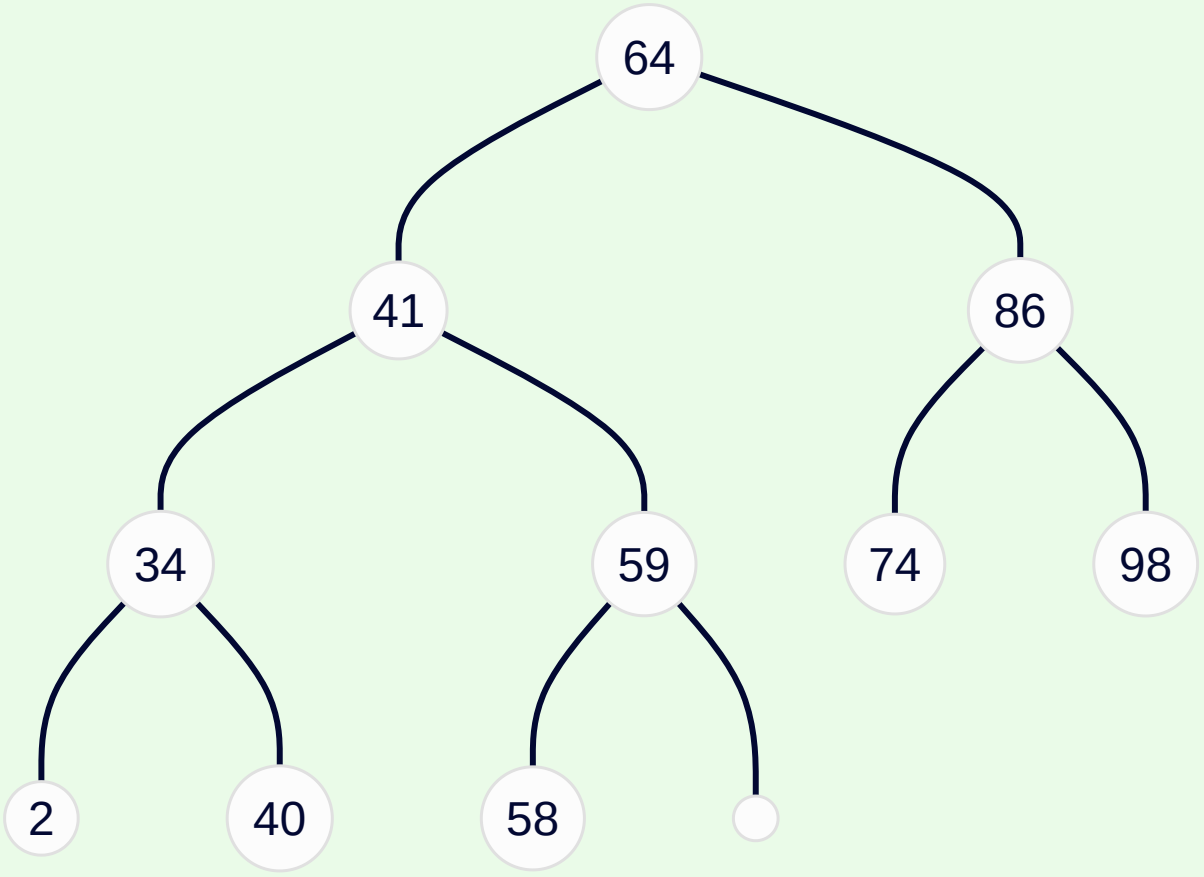

74
34
98
2
64

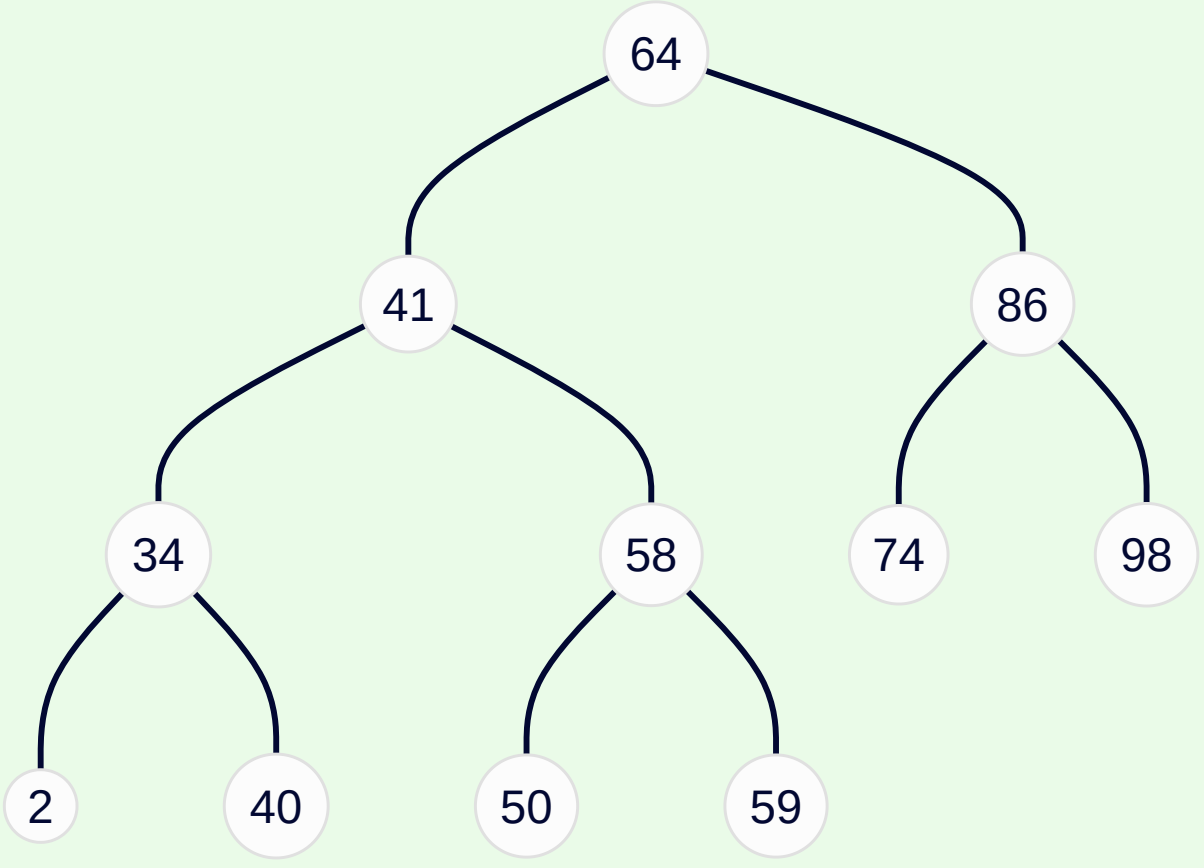

64
34
74
2
41
98

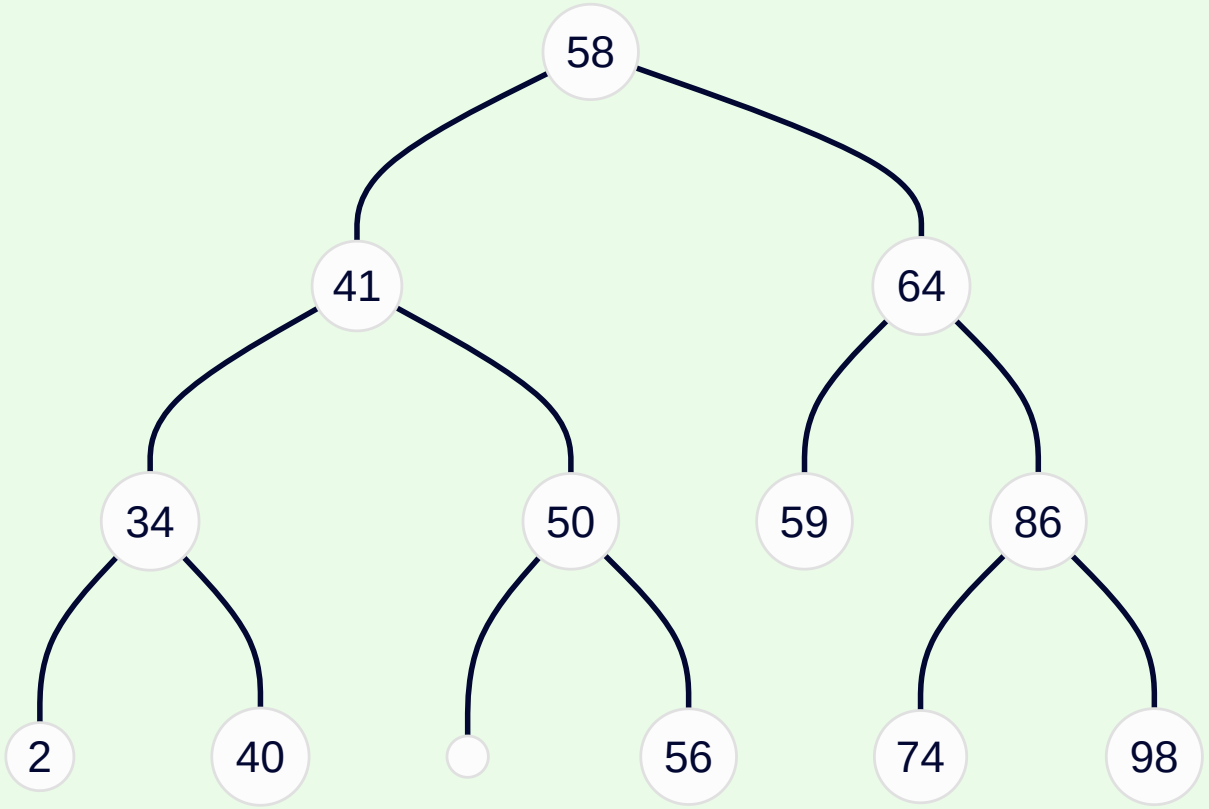

64
34
86
2
41
74
98

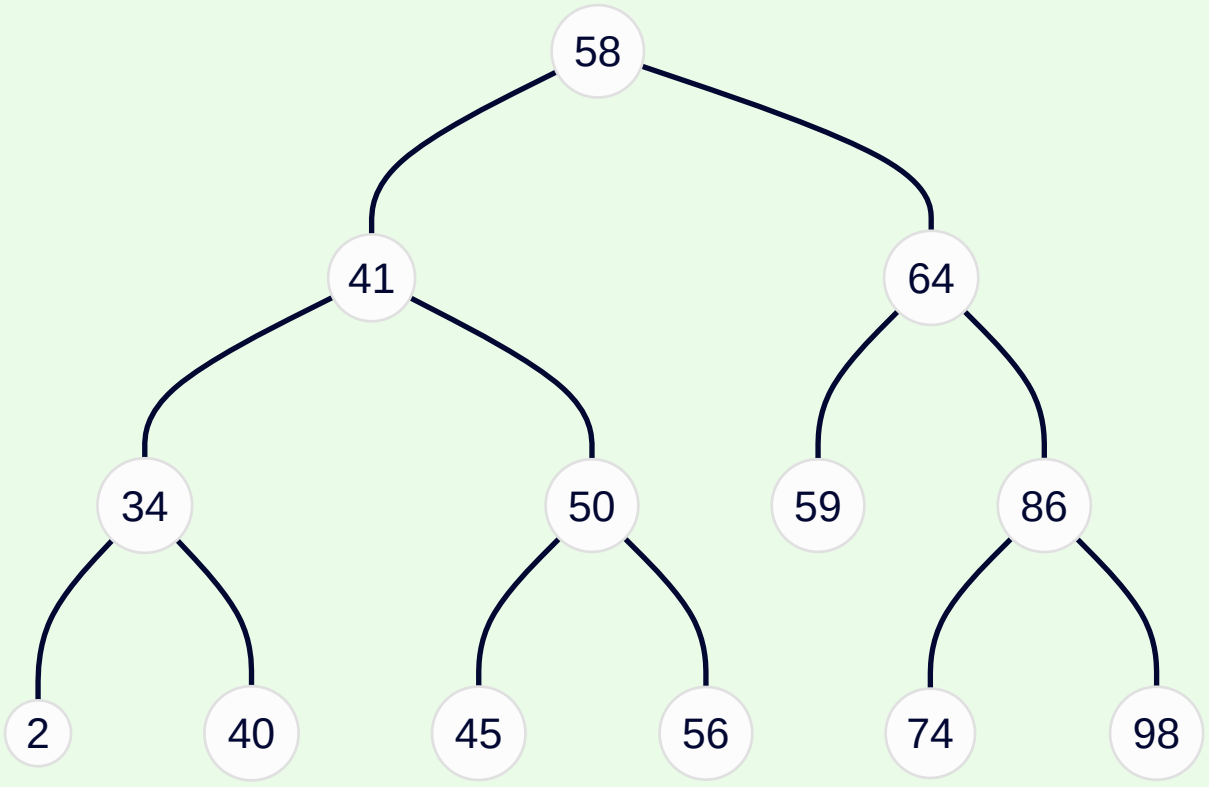

64
34
86
2
41
74
98
59

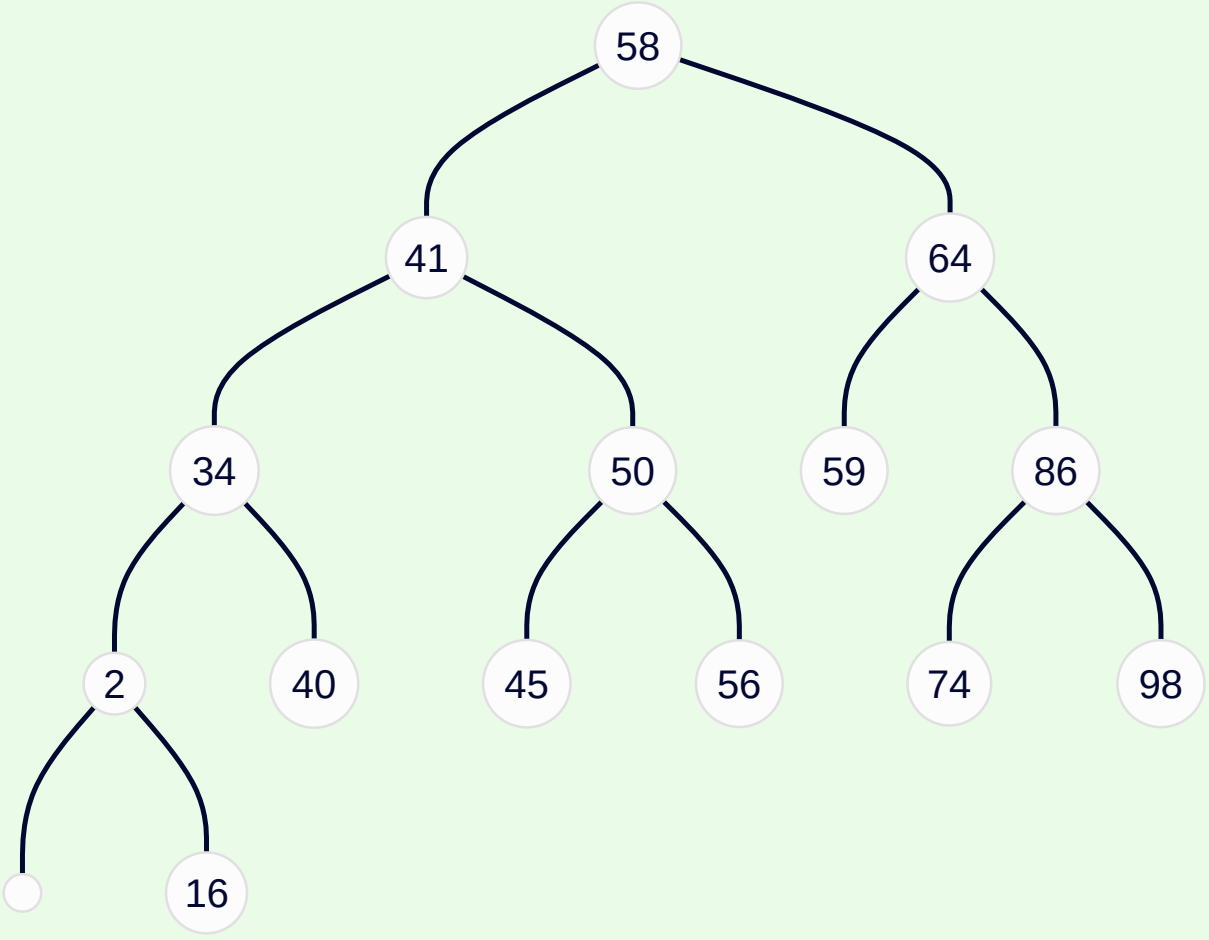

64
34
86
2
41
74
98
40
59

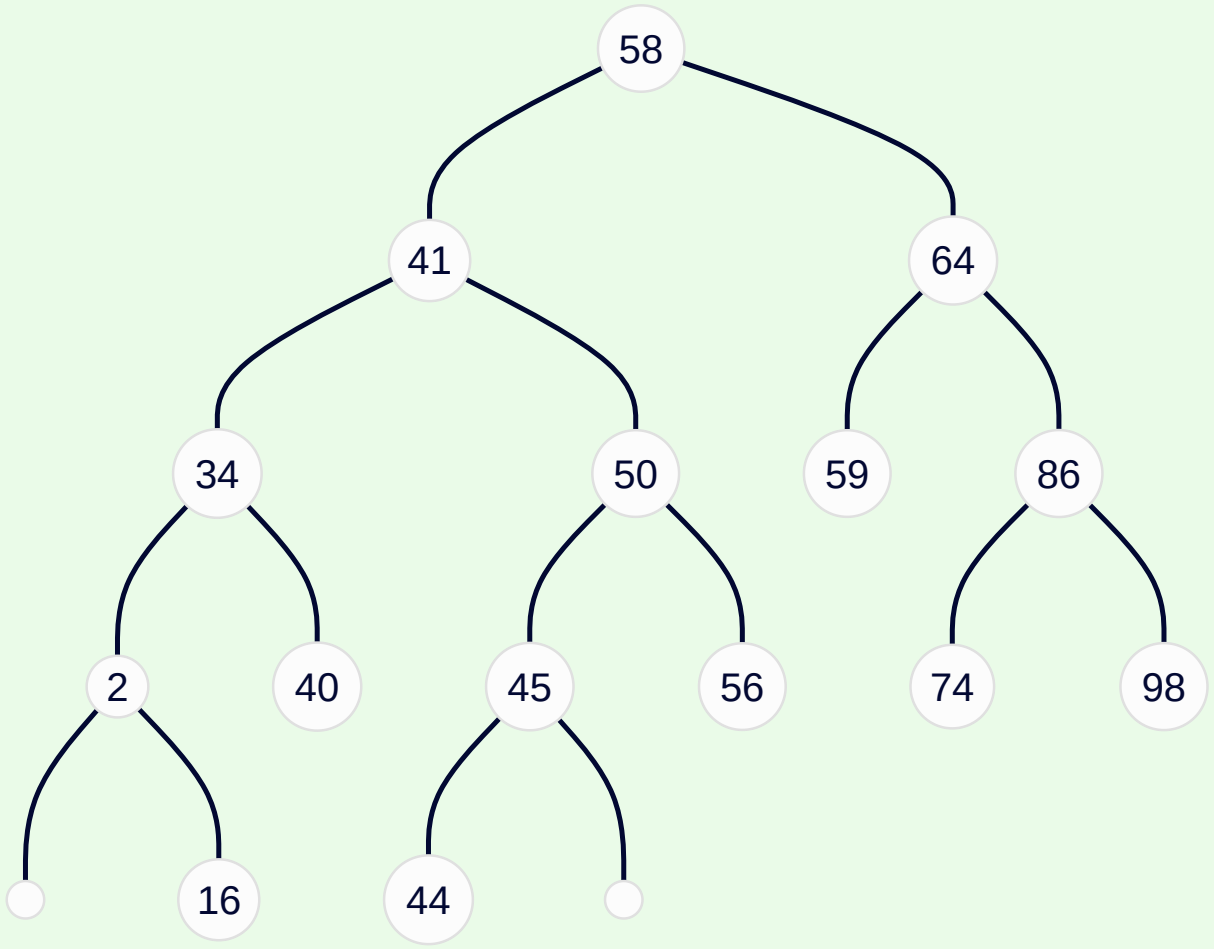

64
41
86
34
59
74
98
2
40
58


64
41
86
34
58
74
98
2
40
50
59


58
41
64
34
50
59
86
2
40
56
74
98


58
41
64
34
50
59
86
2
40
45
56
74
98


58
41
64
34
50
59
86
2
40
45
56
74
98
16

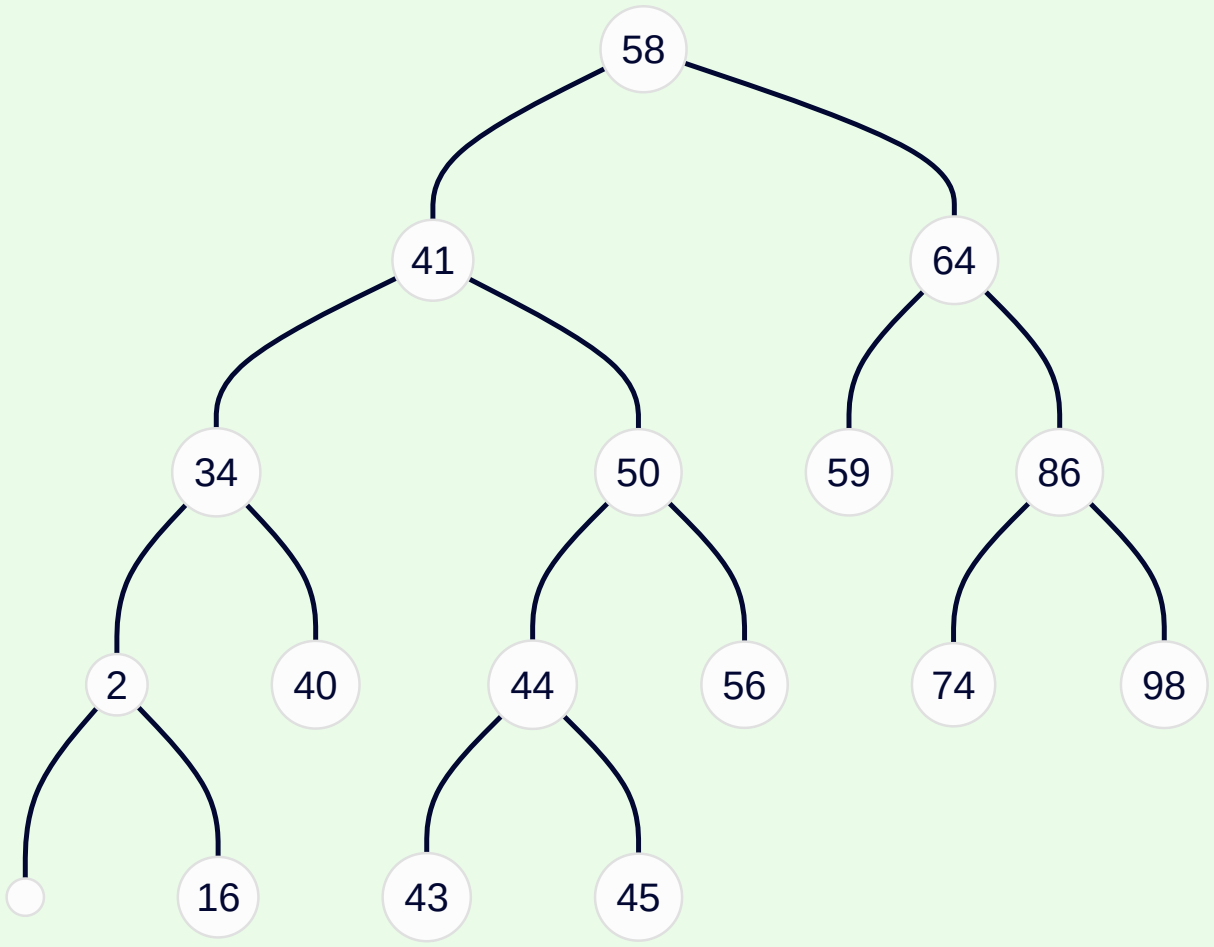


b

Add an element to the AVL tree such that it will cause a right rotation of the tree. State the element and show the tree after. Use the tree from part A.

✓ **Answer**

43

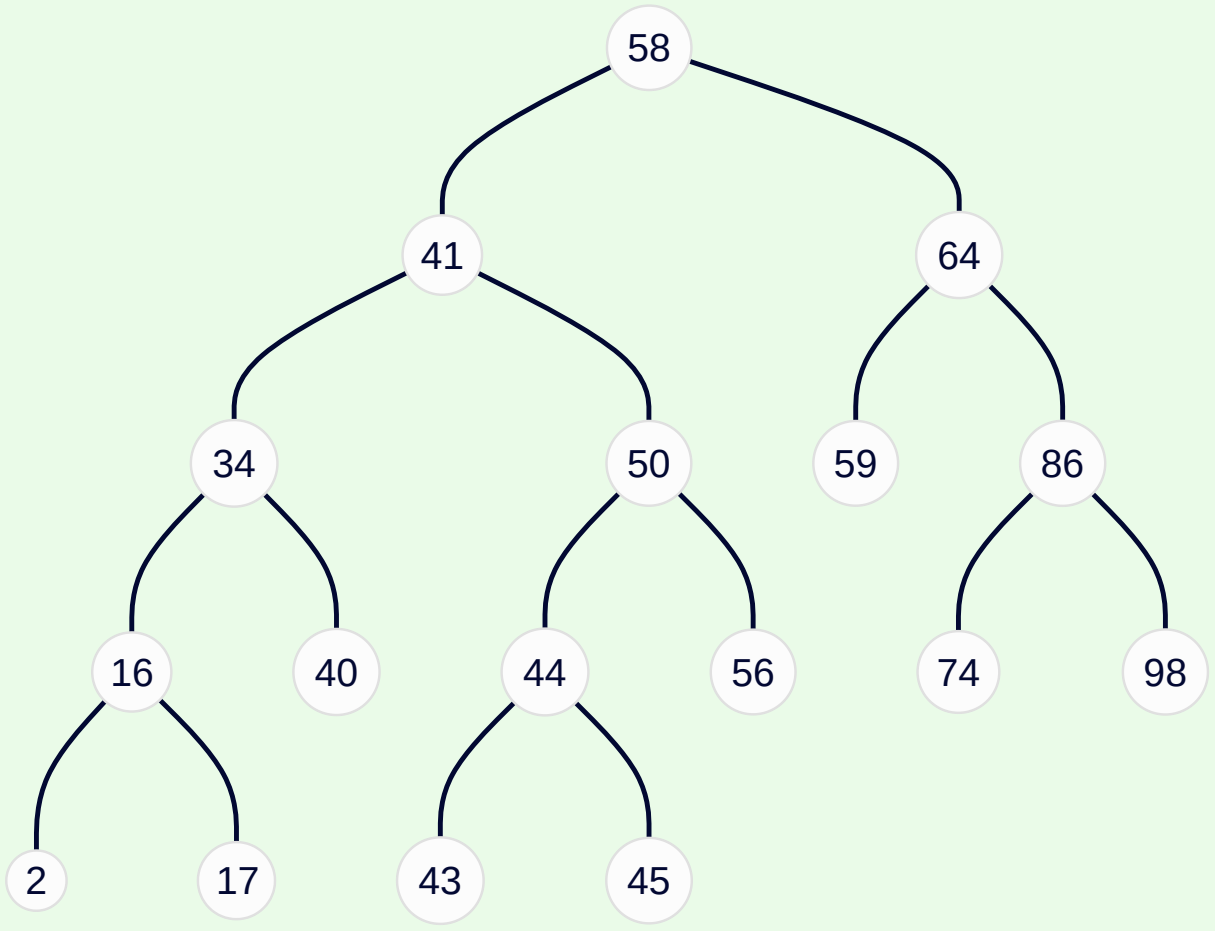


c

Add an element to the AVL tree such that it will cause a left rotation of the tree. State the element and show the tree after. Use the tree from part B.

✓ **Answer**

17



d

Delete an element from the AVL tree such that it will cause a double rotation to the tree. State the element and show the final tree after deletion. Use the tree from part C.

✓ **Answer**

Answer brok :(

2

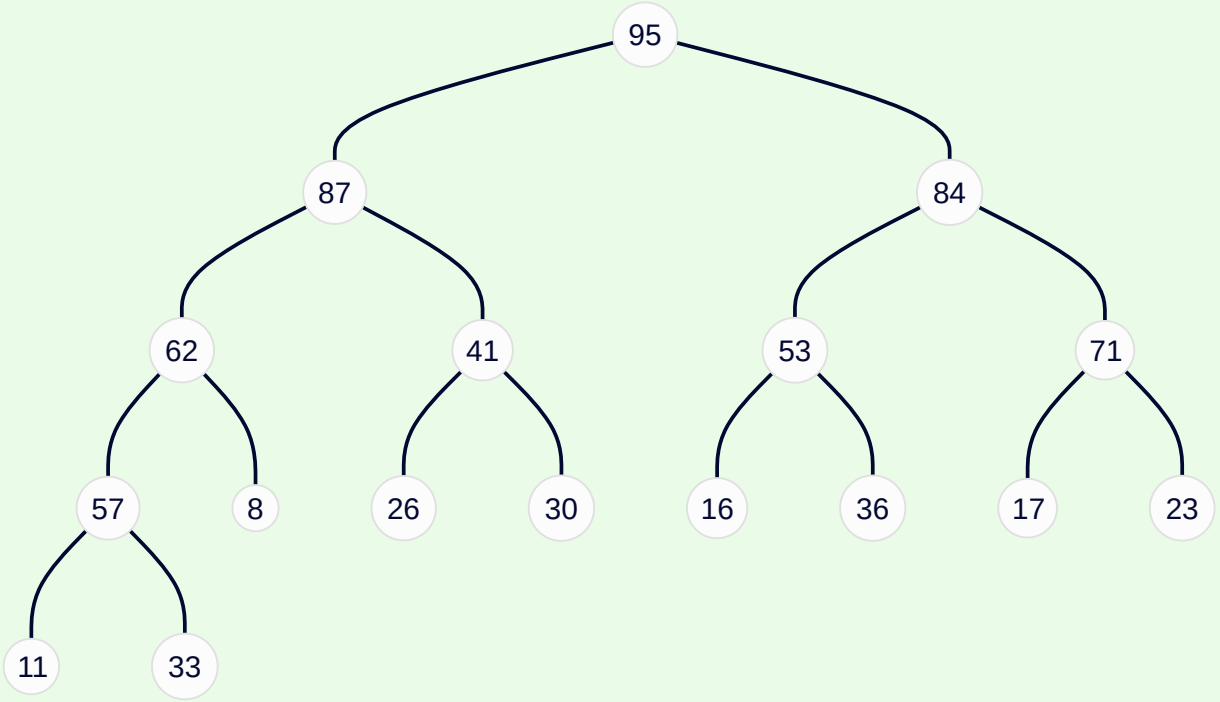
Heap

a

Create a binary max on top heap with the keys [11, 16, 33, 36, 26, 41, 17, 84, 8, 57, 30, 71, 53, 23, 87, 95, 62]. Show the heap after all insertions.

✓ **Answer**

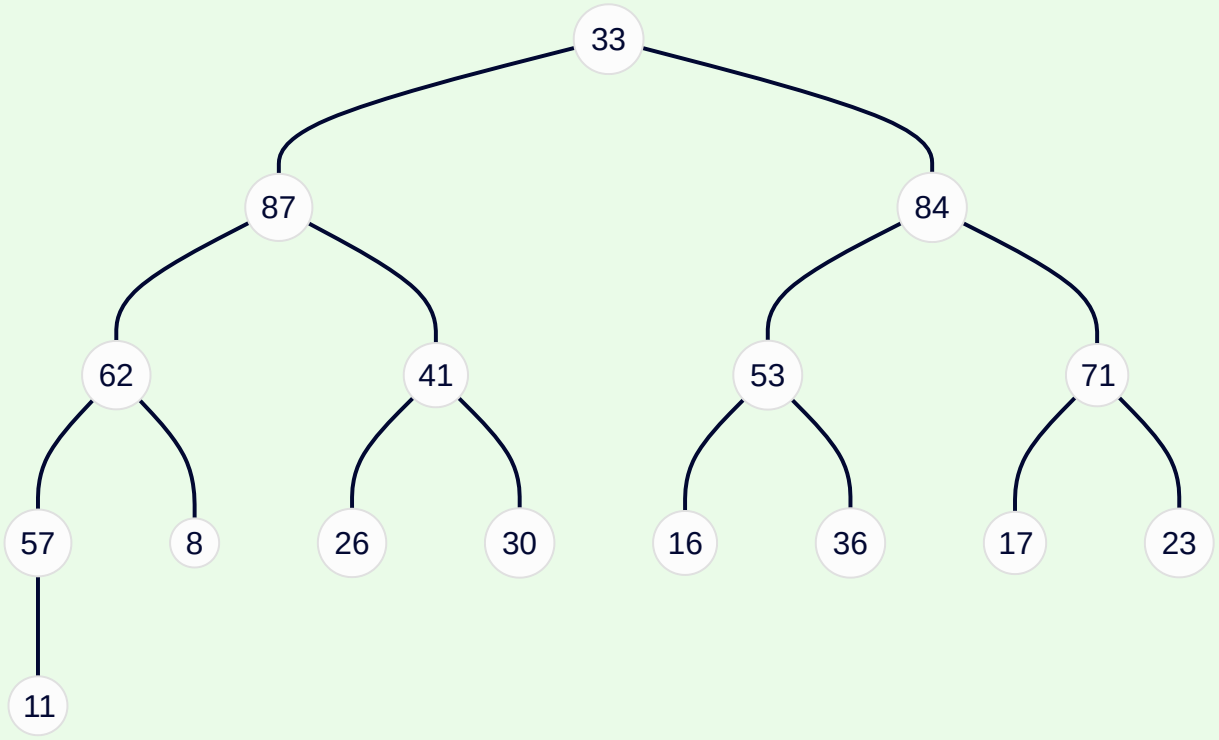
[, 11]
[, 16, 11]
[, 33, 11, 16]
[, 36, 33, 16, 11]
[, 36, 33, 16, 11, 26]
[, 41, 33, 36, 11, 26, 16]
[, 41, 33, 36, 11, 26, 16, 17]
[, 84, 41, 36, 33, 26, 16, 17, 11]
[, 84, 41, 36, 33, 26, 16, 17, 11, 8]
[, 84, 57, 36, 33, 41, 16, 17, 11, 8, 26]
[, 84, 57, 36, 33, 41, 16, 17, 11, 8, 26, 30]
[, 84, 57, 71, 33, 41, 36, 17, 11, 8, 26, 30, 16]
[, 84, 57, 71, 33, 41, 53, 17, 11, 8, 26, 30, 16, 36]
[, 84, 57, 71, 33, 41, 53, 23, 11, 8, 26, 30, 16, 36, 17]
[, 87, 57, 84, 33, 41, 53, 71, 11, 8, 26, 30, 16, 36, 17, 23]
[, 95, 87, 84, 57, 41, 53, 71, 33, 8, 26, 30, 16, 36, 17, 23, 11]
[, 95, 87, 84, 62, 41, 53, 71, 57, 8, 26, 30, 16, 36, 17, 23, 11, 33]

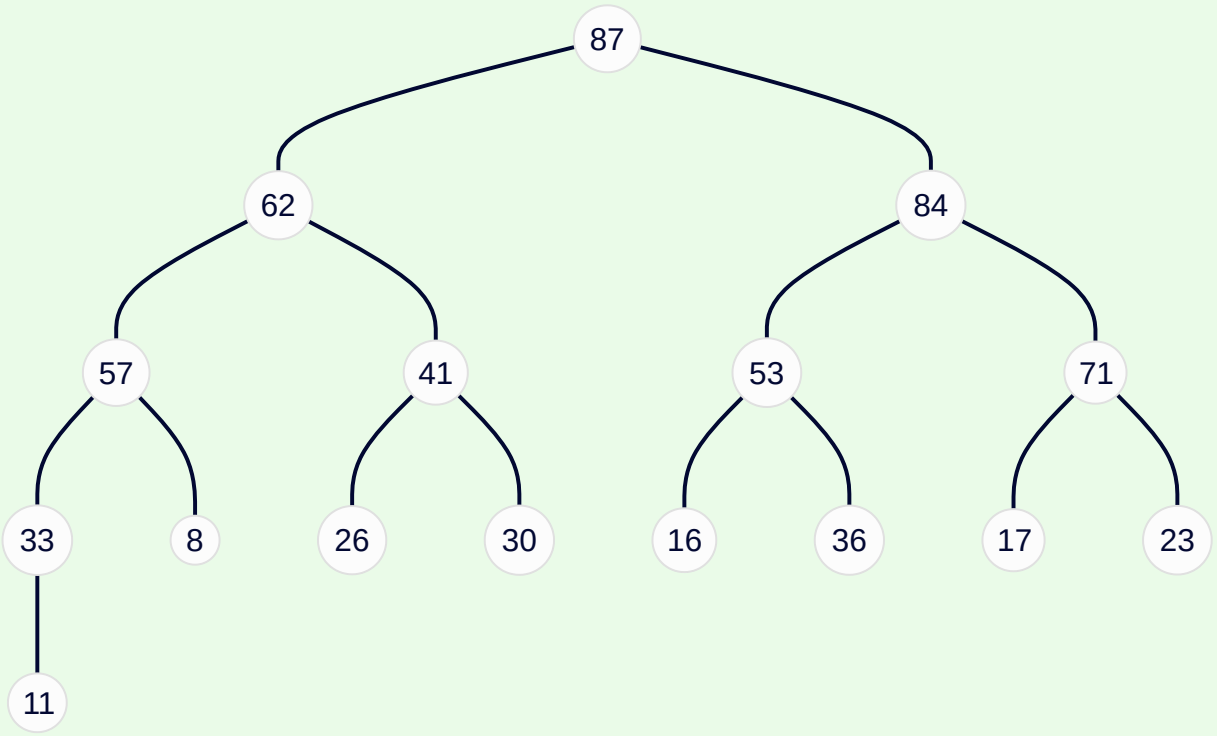


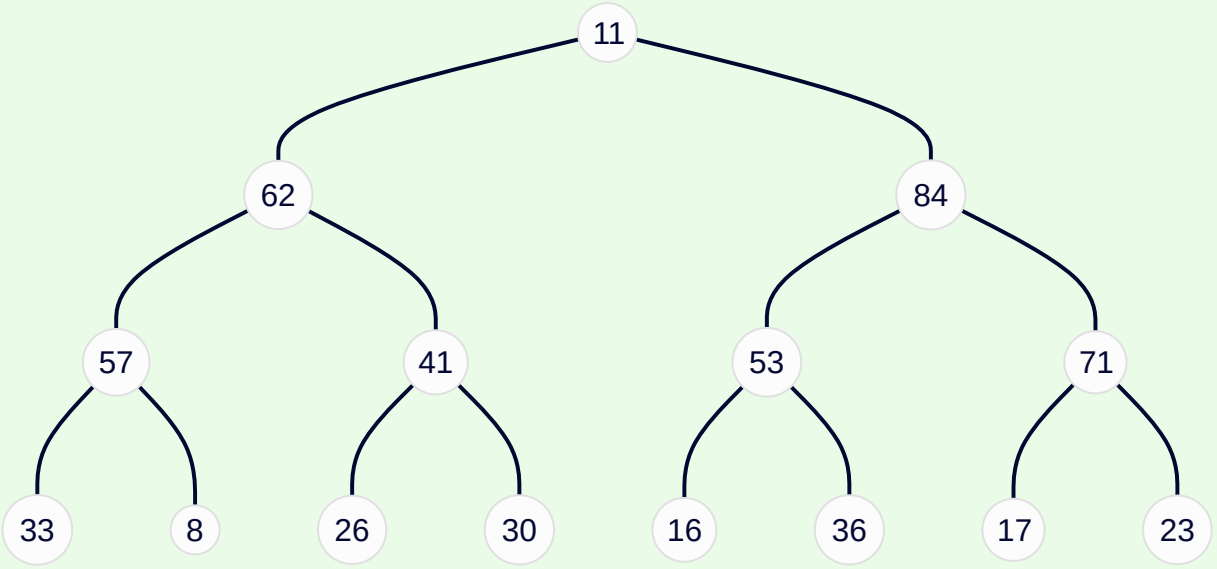
b

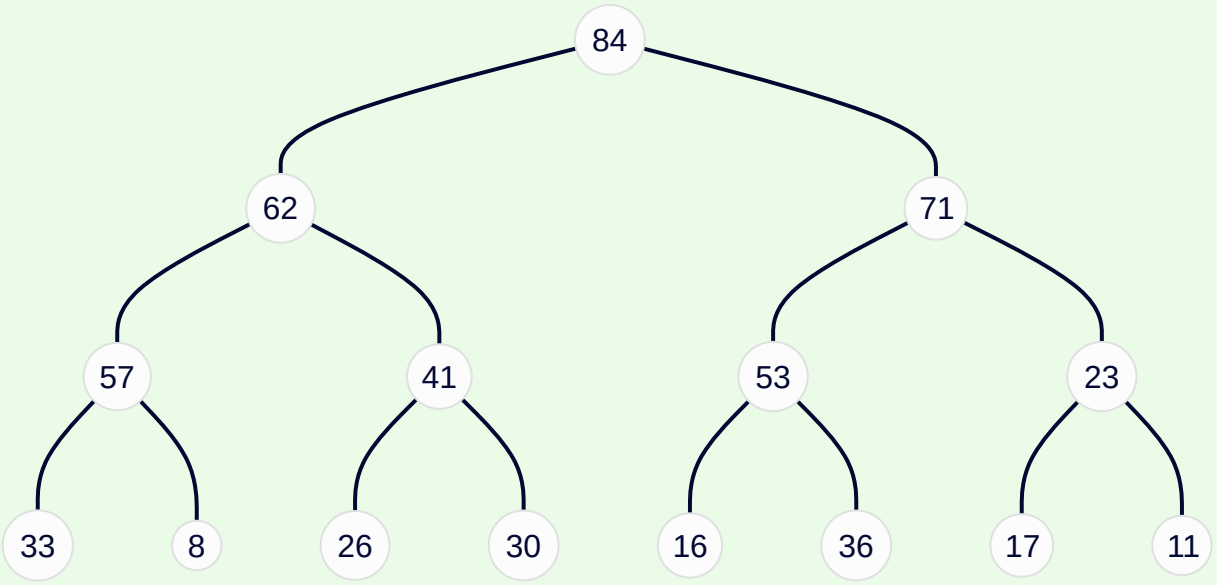
Show the heap when removing the max value 3 times. Show the steps.

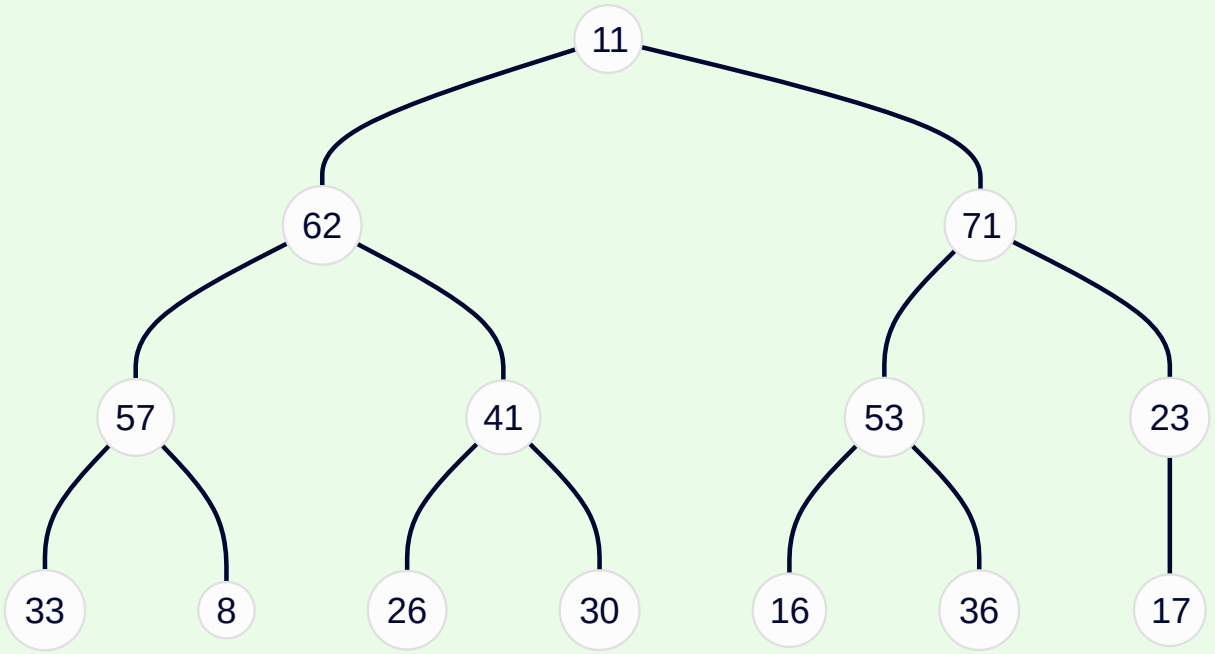
✓ **Answer**

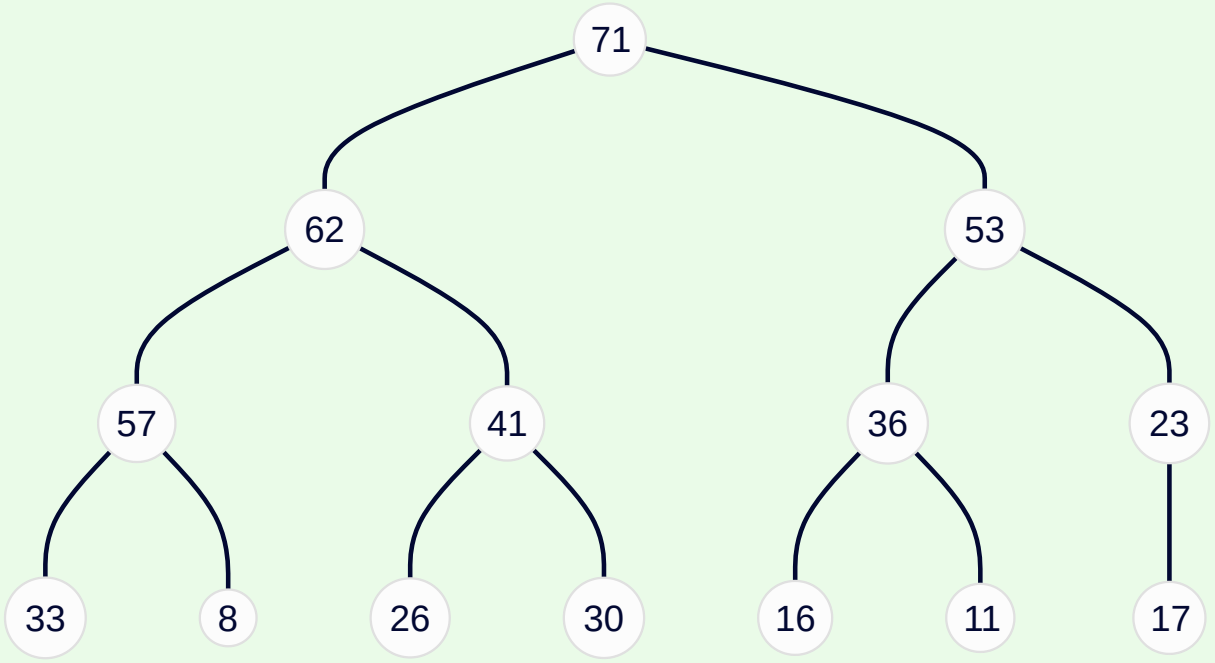



87
62
84
57
41
53
71
33
8
26
30
16
36
17
23
11


11
62
84
57
41
53
71
33
8
26
30
16
36
17
23

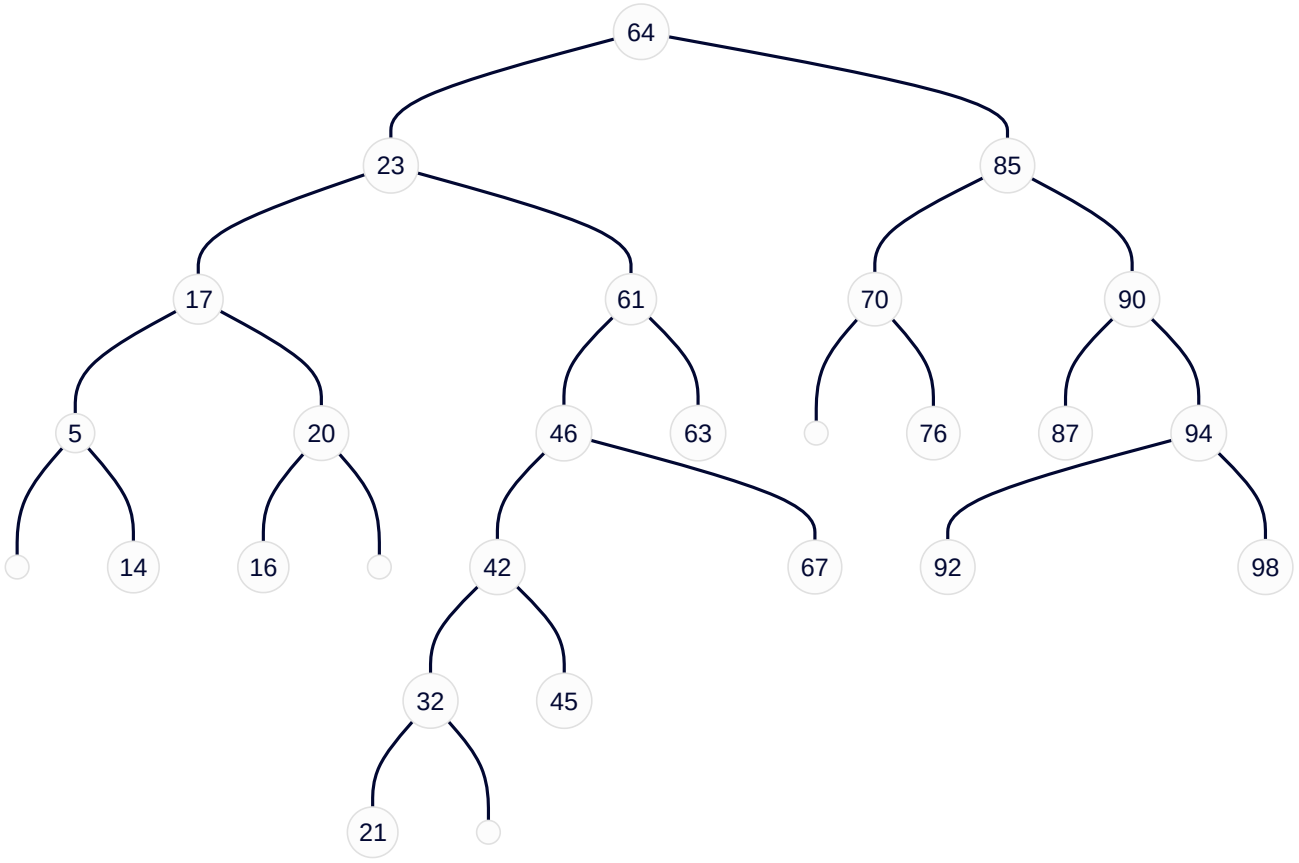

84
62
71
57
41
53
23
33
8
26
30
16
36
17
11


11
62
71
57
41
53
23
33
8
26
30
16
36
17


71
62
53
57
41
36
23
33
8
26
30
16
11
17

3

Given the following tree:



a

Give post-order, pre-order of the tree.

✓ **Answer**

Post-order:
[14, 5, 16, 20, 17, 21, 32, 45, 42, 67, 46, 63, 61, 23, 76, 70, 87, 92, 98, 94, 90, 85, 64]

Pre-order:
[64, 23, 17, 5, 14, 20, 16, 61, 46, 42, 32, 21, 45, 67, 63, 85, 70, 76, 90, 87, 94, 92, 98]

b

Give level-order and in-order of the tree

✓ **Answer**

Level-order:
[64, 23, 85, 17, 6, 70, 90, 5, 20, 46, 63, 76, 87, 94, 14, 16, 42, 67, 92, 98, 32, 45, 21]

In-order:
[5, 14, 17, 16, 20, 23, 21, 32, 42, 45, 46, 67, 61, 63, 64, 70, 76, 85, 87, 90, 92, 94, 98]

c

When looking for the value 67, give the path traced by breadth first search and depth first search algorithm

✓ **Answer**

Breadth first/Level-order:
[64, 23, 85, 17, 6, 70, 90, 5, 20, 46, 63, 76, 87, 94, 14, 16, 42, 67, 92, 98, 32, 45, 21]

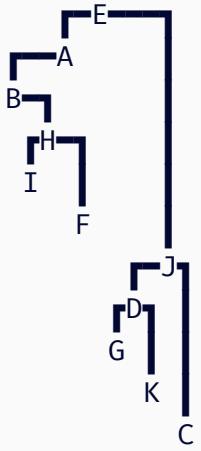
Depth first/pre-order/post-order (depending on implementation):
[64, 23, 17, 5, 14, 20, 16, 61, 46, 42, 32, 21, 45, 67, 63, 85, 70, 76, 90, 87, 94, 92, 98]
[14, 5, 16, 20, 17, 21, 32, 45, 42, 67, 46, 63, 61, 23, 76, 70, 87, 92, 98, 94, 90, 85, 64]

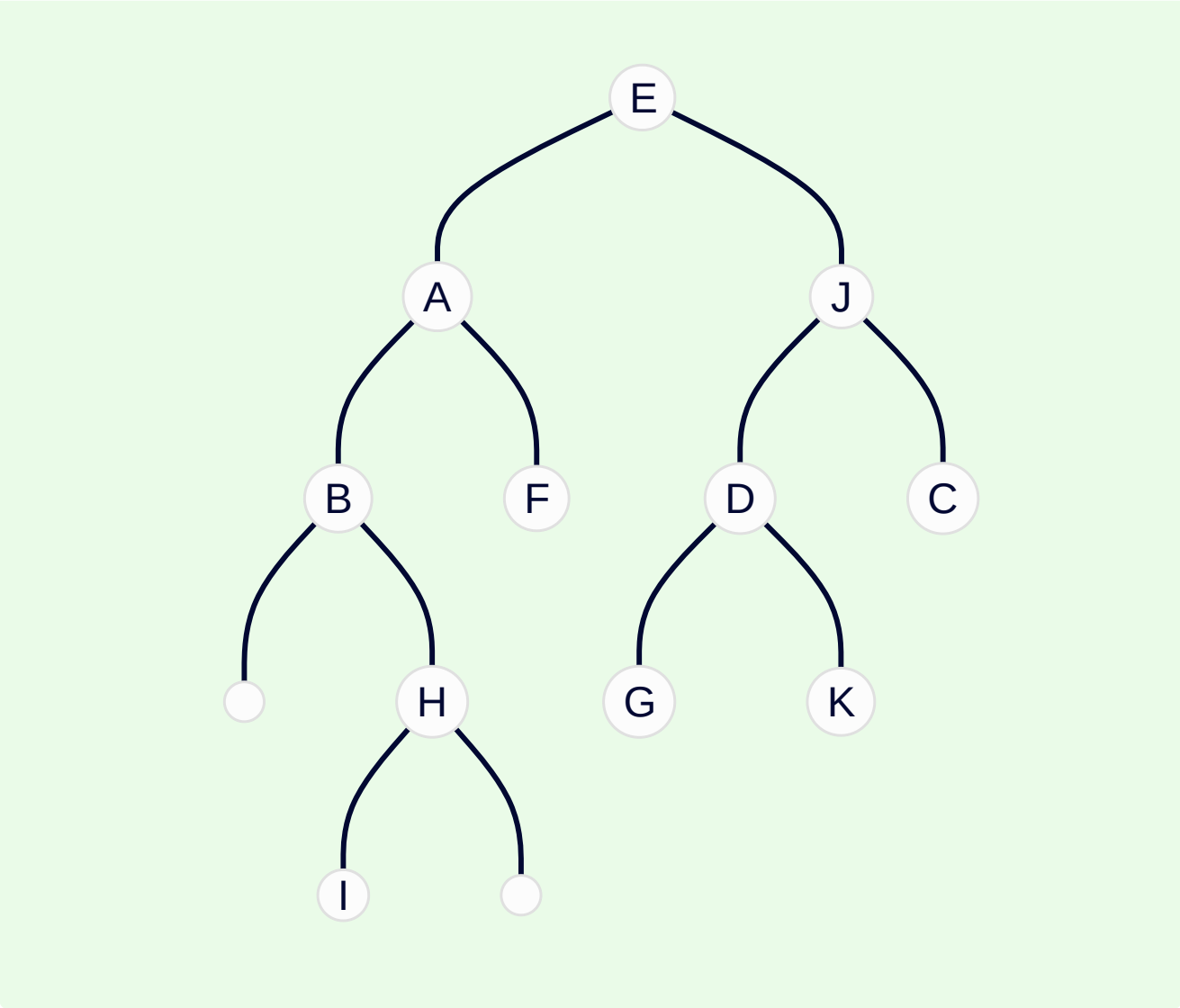
d

Given a tree with the
preorder: [E, A, B, H, I, F, J, D, G, K, C] and the
inorder: [B, I, H, A, F, E, G, D, K, J, C], draw the binary tree.

✓ **Answer**

L<-->R is Inorder
U<-->D is Preorder




E
A
J
B
F
D
C
H
G
K
I