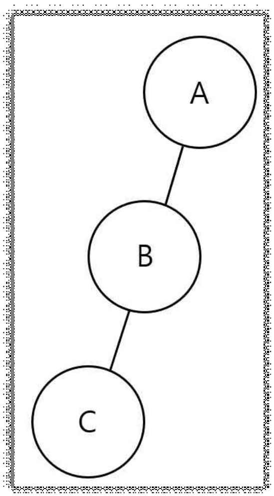
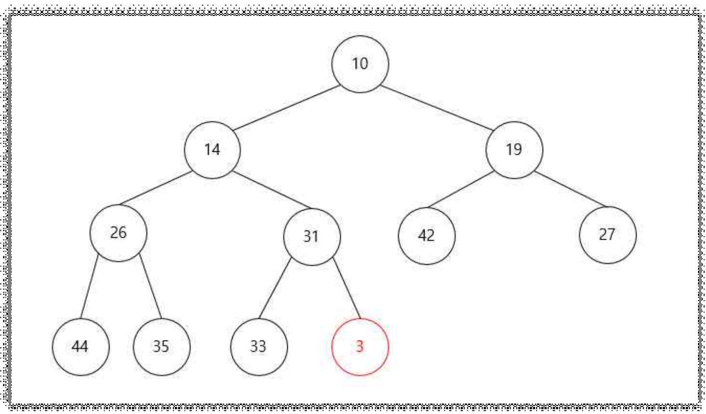
1.
Pre-order traversal : A-B-D-G-I-C-E-H-J-K-F
Post-order traversal : I-G-D-B-J-K-H-E-F-C-A

2.
Pre-order traversal can't be same order with Post-order traversal. Since a tree have more than one node, there must at least two nodes, root node and its child node. So unless a tree has just one node, pre-order and post-order traversal can't be same. In pre-order traversal, the first visited node is root node. But in post-order traversal, the first visited node is one of root node's child, not root node. So two traversals can't be same.

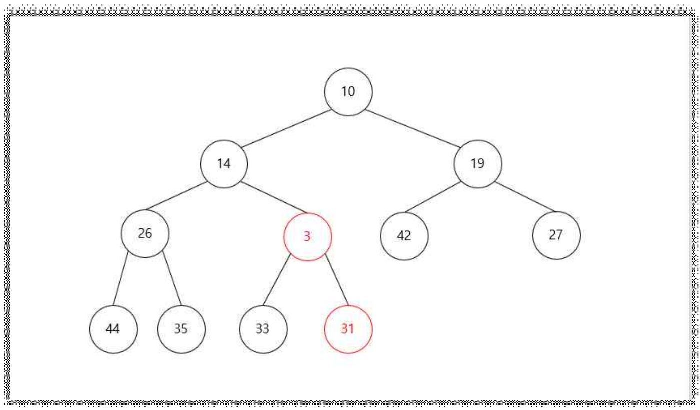


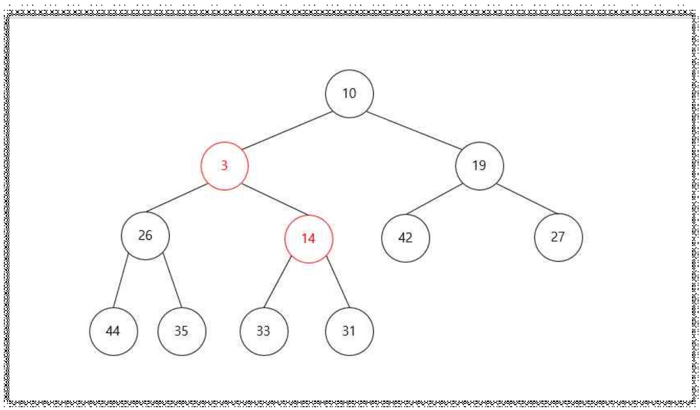
It is possible. Like a tree in the left picture, pre-order traversal is A-B-C which is the reverse order of post-order traversal, C-B-A.

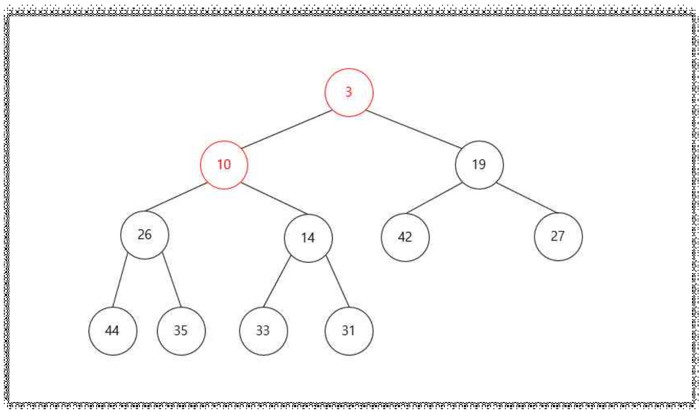
3.
- Insertion
① Find the insertion node, the new last node. Store value 3 at the last node.



② To restore the heap-order priority, upheap the new value 3 until it reaches root or node whose parent has a key smaller than or equal to 3.

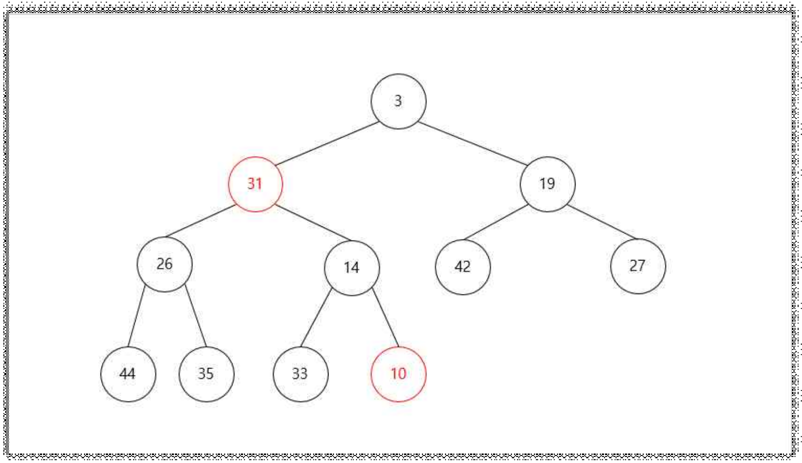




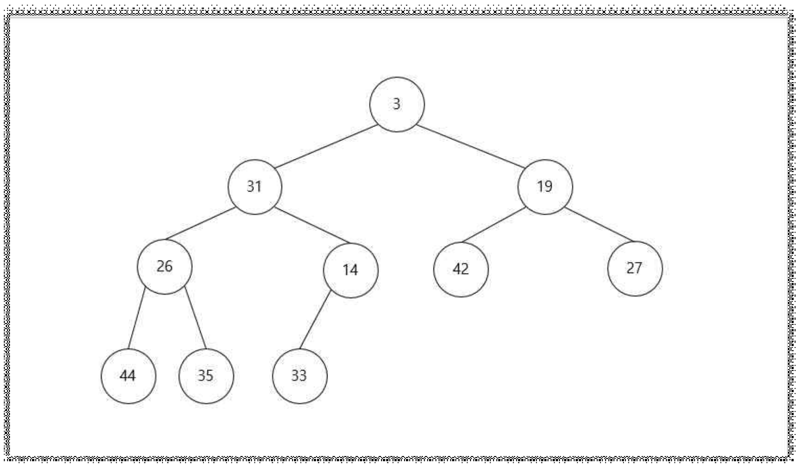


- Deletion

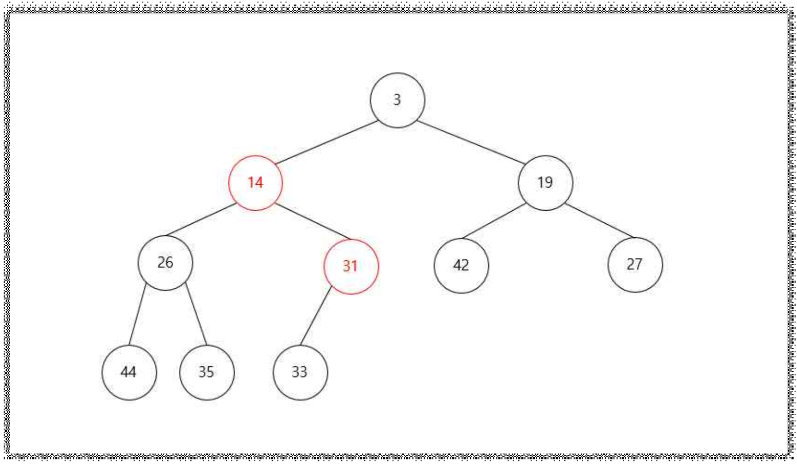
① Swap the node with value 10 and the last node.



② Remove the last node.



② To restore heap-order priority, downheap the value 31 until the value reaches a leaf or a node whose children have keys greater than or equal to 31.



4.

(a) If a=5, b=0, N=6 then $(x) = 5x(\mathrm{mod}6)$.

$h(0) = 0, h(1) = 5, h(2) = 4, h(3) = 3, h(4) = 2, h(5) = 1$.

So, $\Pr(h(c) = d) = \dfrac{1}{6}$

(b) If $a = 4, b = 0, \quad = 6$ then $h(x) = 4x(\mathrm{mod}6)$. In this case,

$h(0) = 0, h(1) = 4, h(2) = 2, h(3) = 0, h(4) = 4, h(5) = 2$

So, $\Pr(h(c) = 0) = \Pr(h(c) = 2) = \Pr(h(c) = 4) = \dfrac{1}{3}$,

$\Pr(h(c) = 1) = \Pr(h(c) = 3) = \Pr(h(c) = 5) = 0$

Given probability does not hold.

5.

a)

First, perform $Insert(15)$.

$h\ (15) = 2, h_2(15) = 7, h_3(15) = 8$, so bloom filter array will be

0	0	1	0	0	0	0	1	1	0	0

Second, perform $Insert(1000)$.

$h_1(1000) = 9, h_2(1000) = 10, h_3(1000) = 2$, so bloom filter array will be

0	0	1	0	0	0	0	1	1	1	1

Third, perform $sert(22)$.
$h\ (22)=1, h_2(22)=5, h_3(22)=1$, so bloom filter array will be

0	1	1	0	0	1	0	1	1	1	1

Fourth, perform $Insert(5)$.
$h_1(5)=5, h_2(5)=2, h_3(5)=7$, so bloom filter array will be

0	1	1	0	0	1	0	1	1	1	1

b) Like the case of question (a), let assume that we perform $Insert(15)$, $Insert(1000)$, $Insert(22)$. Then array will be

0	1	1	0	0	1	0	1	1	1	1

After insertion, do $Contains(5)$. Because of the hash values are $h_1(5)=5, h_2(5)=2$, $h_3(5)=7$ and corresponding array values are $w[5]=w[2]=w[7]=1$(w is array), it returns $true$. But value 5 has not been inserted.

c)
To return $true$ for all positive integer, all of the entries would be filled by 1. Suppose the series of following instruction are performed.
$Insert(0), Insert(1), Insert(2), Insert(3), Insert(4), Insert(8)$

The hash values of those instructions are
$h_1(0)=1, h_2(0)=5, h_3(0)=1$
$h_1(1)=4, h_2(1)=0, h_3(1)=0$
$h_1(2)=7, h_2(2)=6, h_3(2)=10$
$h_1(3)=10, h_2(3)=2, h_3(3)=9$
$h_1(4)=2, h_2(4)=7, h_3(4)=8$
$h_1(8)=3, h_2(8)=7, h_3(8)=4$
So, array will be

1	1	1	1	1	1	1	1	1	1	1

After those insertion, bloom filter returns $true$ for all positive integer.