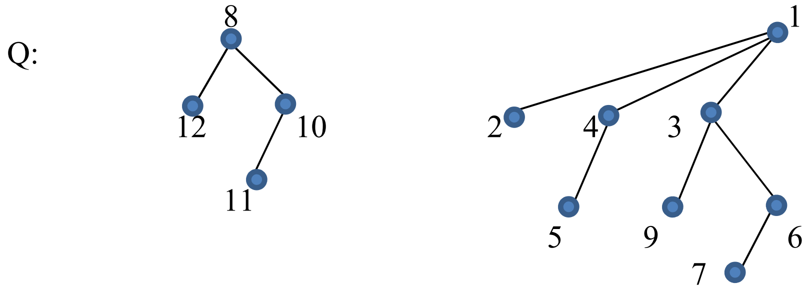
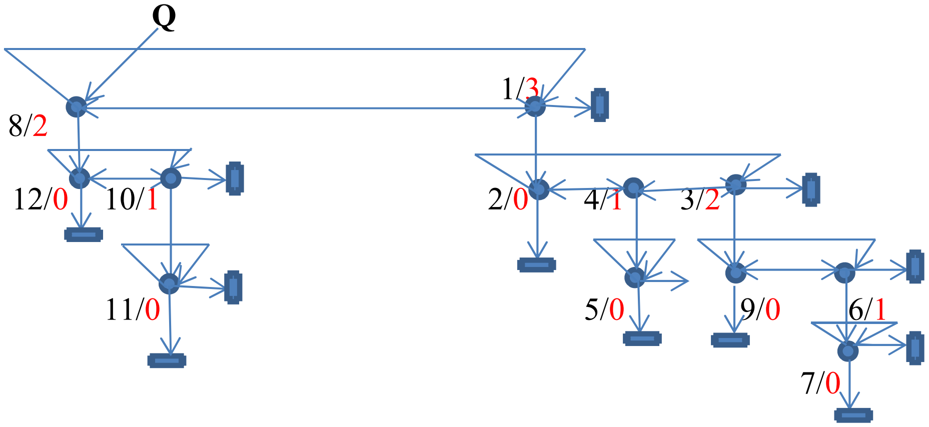
Instruction: You must show your tree clearly after each insertion/deletion/modification for credit. No credit will be given if you do not show all your trees/work.

1. (15) Construct a Binomial Queue Q by inserting <2, 1, 4, 5, 9, 3, 6, 7, 8, 12, 11, 10>, in the given order, into an initially empty queue. When done, illustrate the data structure used in implementing the resulting Binomial queue.
   Solution:

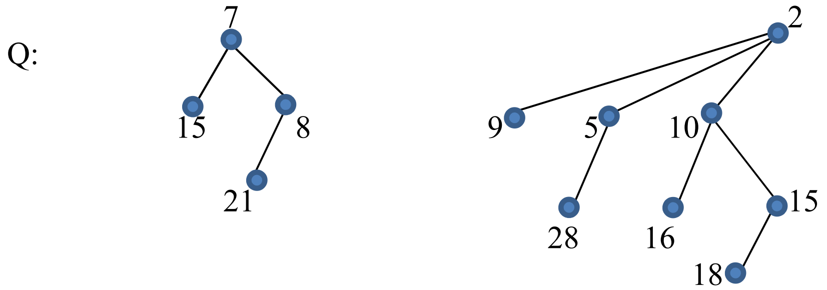
   Final BQ:



Data structure for Q:

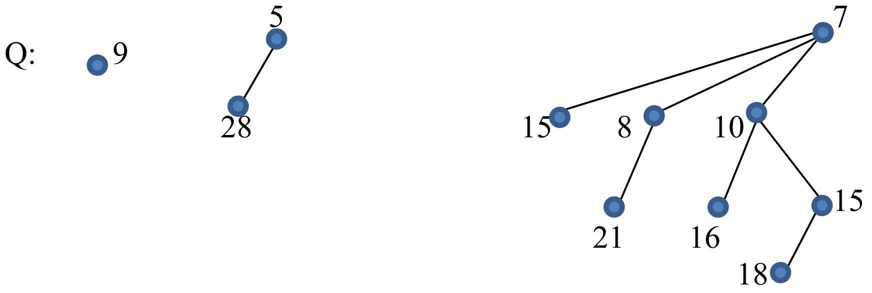


2. (15) Perform deleteMin(Q) operation on the following Binomial Queue Q. When done, illustrate the data structure used in implementing the resulting binomial queue.

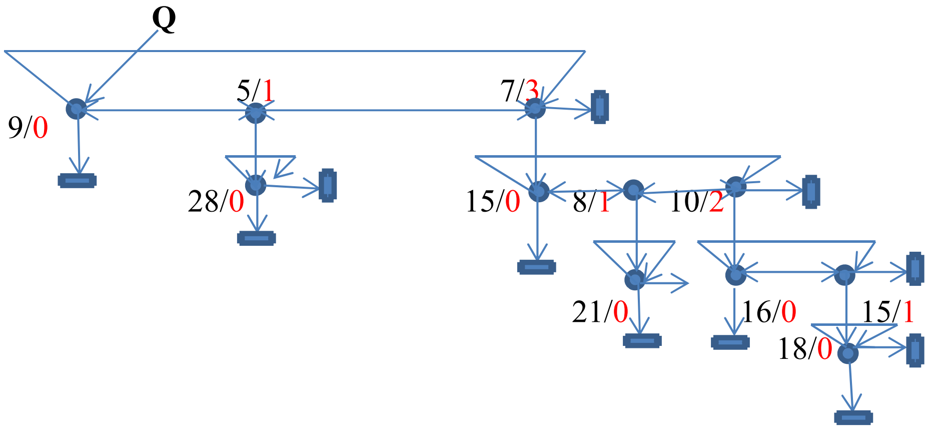


Solution:

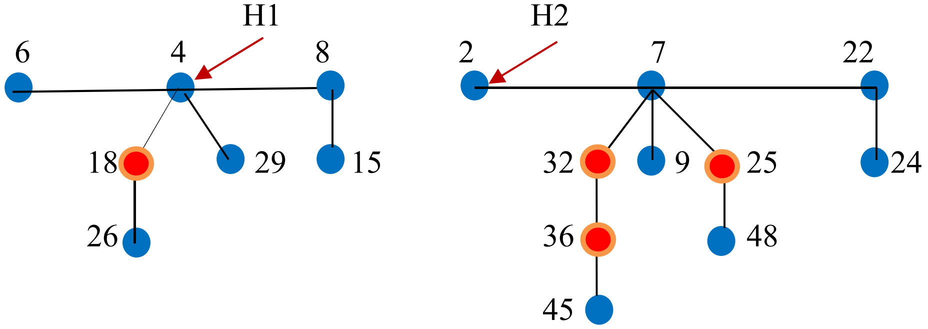
deleteMin(Q):



Data structure:

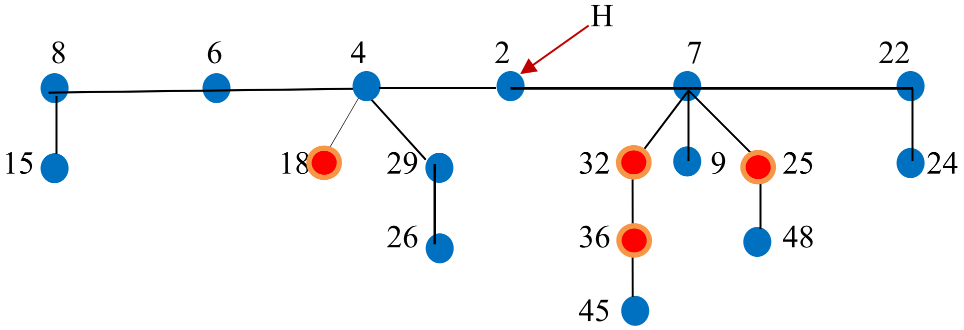


3. (10) Perform concate(H1,H2) on the following two Fibonacci heaps H1 and H2, where 🔴 is a marked node.

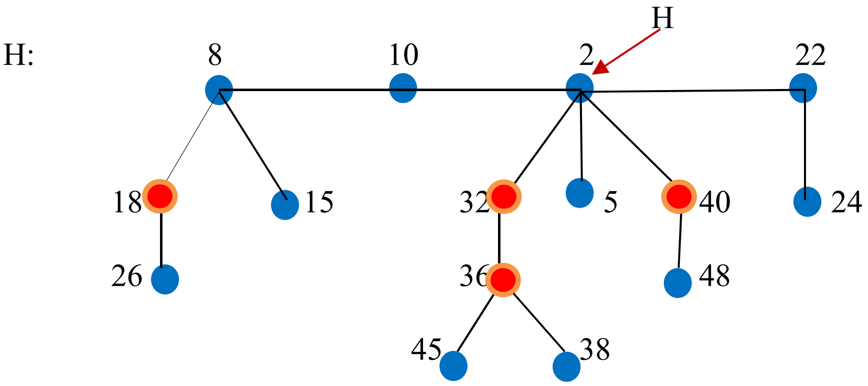


Solution:

concate(H1,H2):

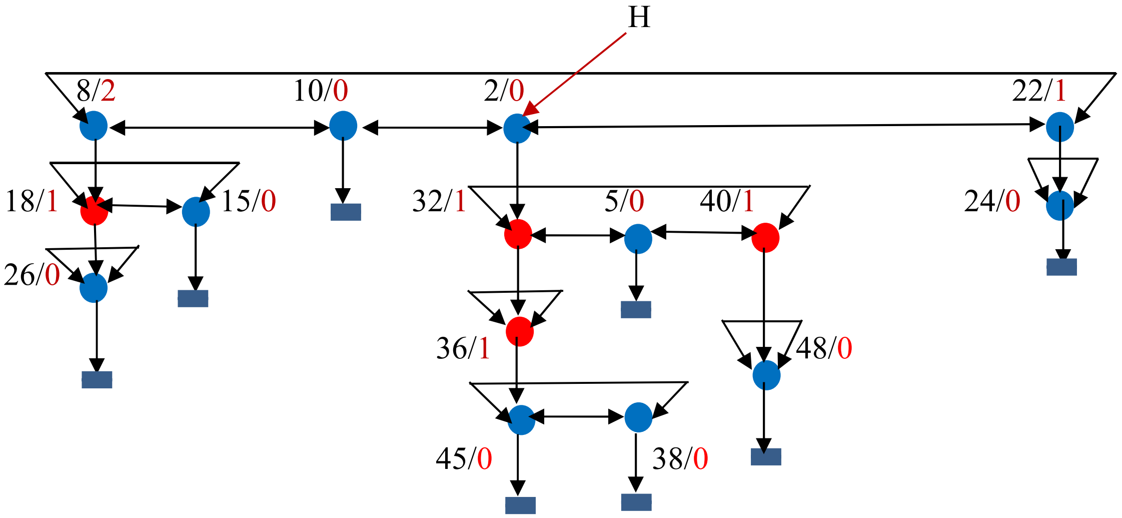


4. (10) Show the data structure used in implementing the following Fibonacci heap H, where ● is a marked node.



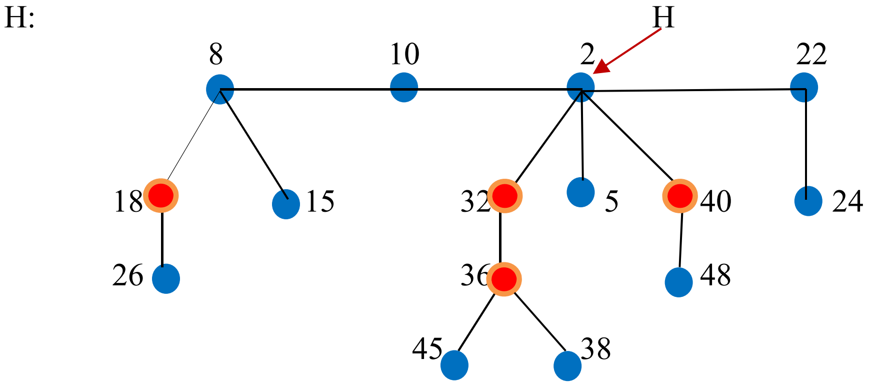
Solution:

Implementation of H with degree information:



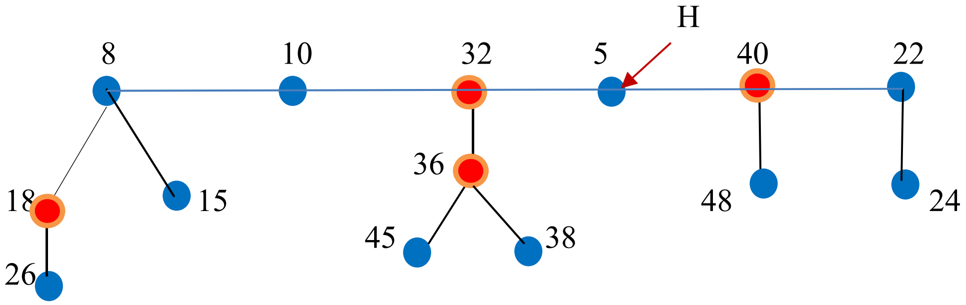
Remark: Each node will have a parent pointer pointing to its parent. For simplicity, the parent pointer of each node is omitted.

5. (15) Perform deleteMin(H) on the following Fibonacci heap H, where ● is a marked node. *Remark:* you must show all steps clearly for credit.

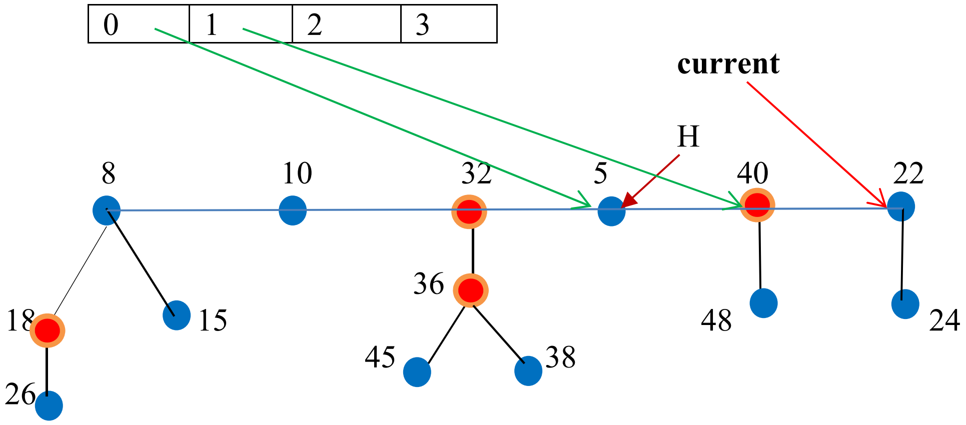


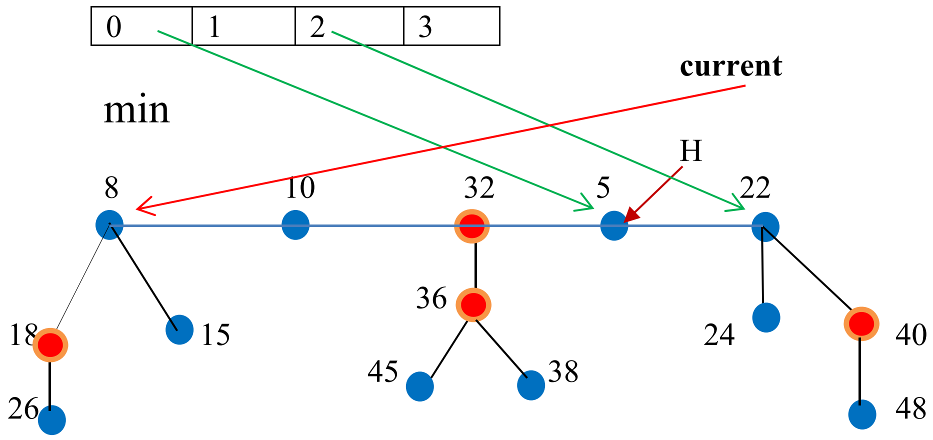
Solution:

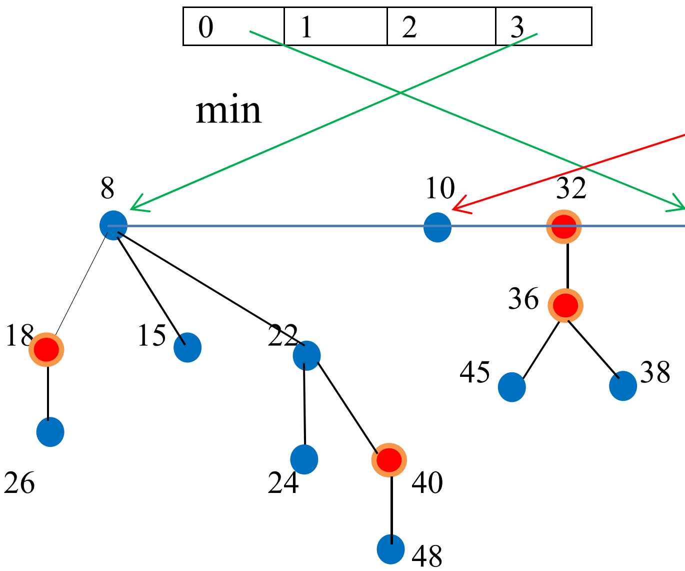
deleteMin(H):

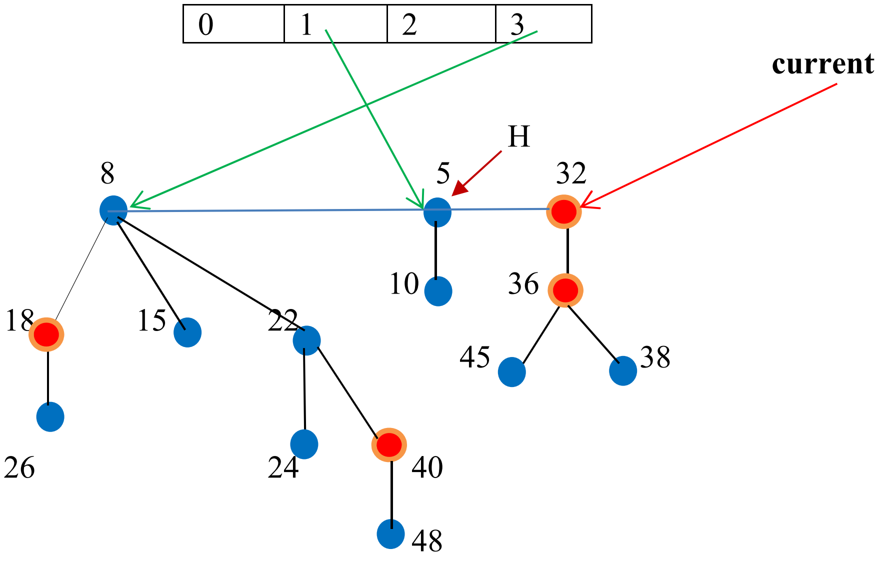


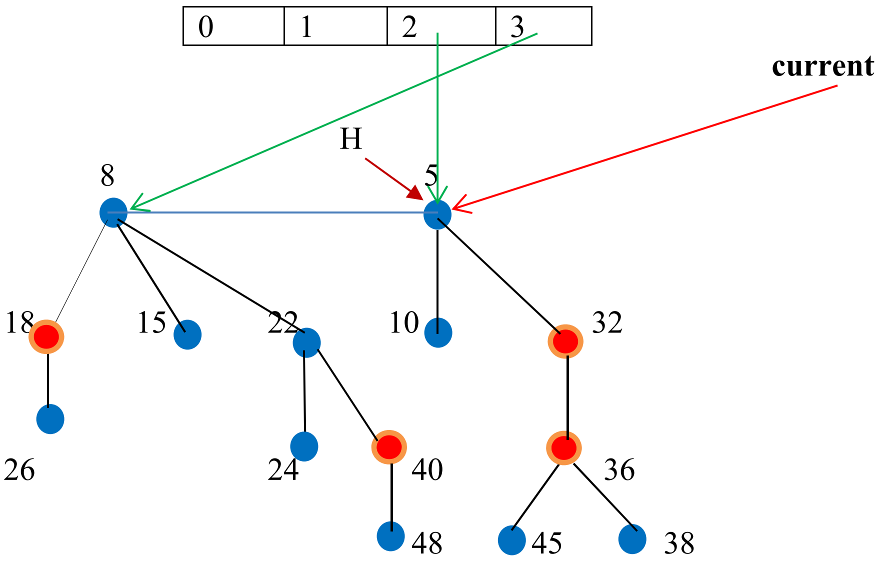
Trees consolidation:



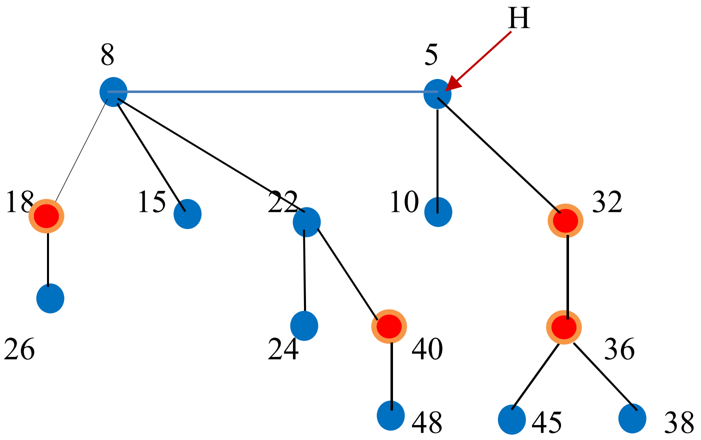

0
1
2
3
current
min
H
8
10
32
5
22
36
18
15
24
40
45
38
26
48


0
1
2
3
current
min
H
8
10
32
5
36
18
15
22
45
38
26
24
40
48

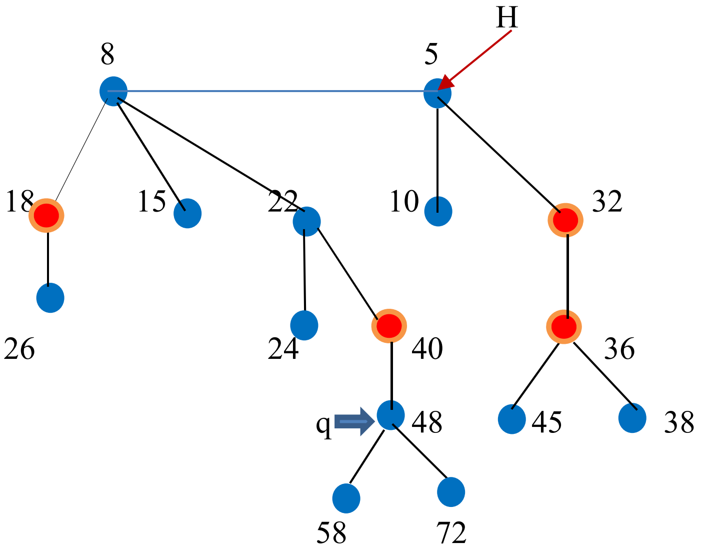

0
1
2
3
current
H
8
5
32
10
36
18
15
22
45
38
26
24
40
48



Final Fibonacci heap:

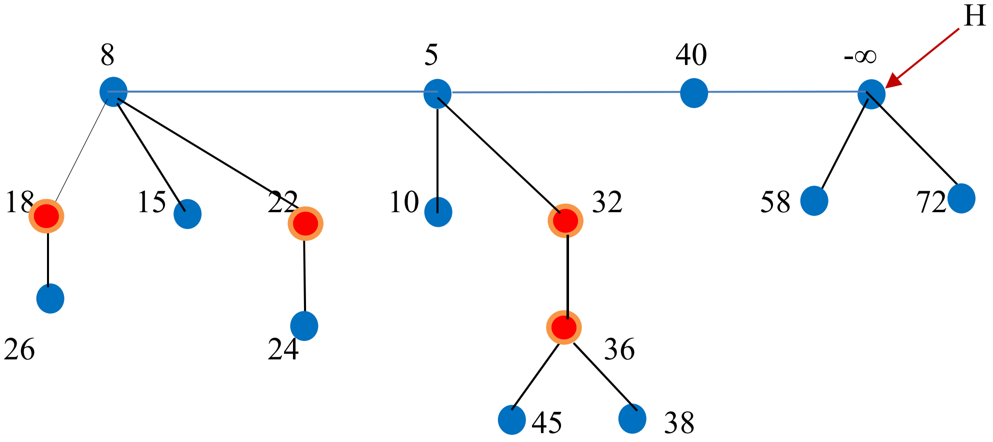


6. (15) Perform delete(q,H) on the following Fibonacci heap H, where 🔴 is a marked node. *Remark:* you must show all steps clearly for credit.

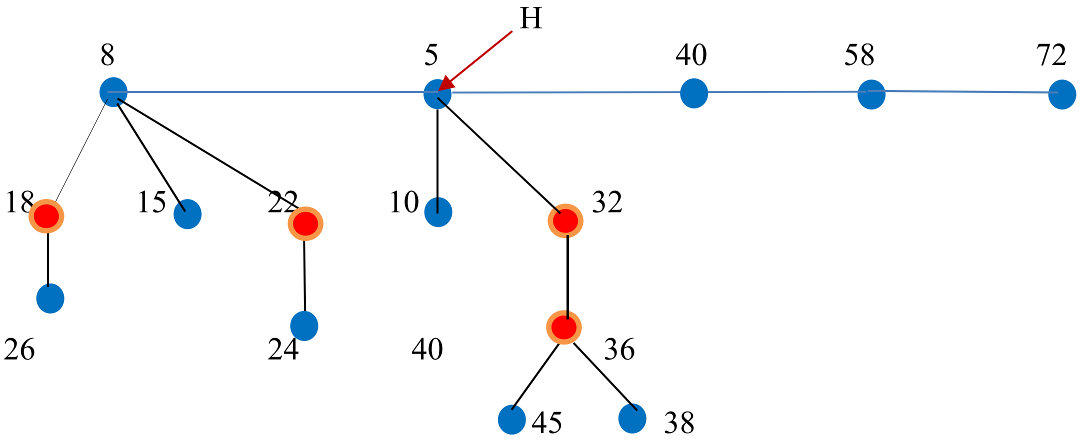


Solution:

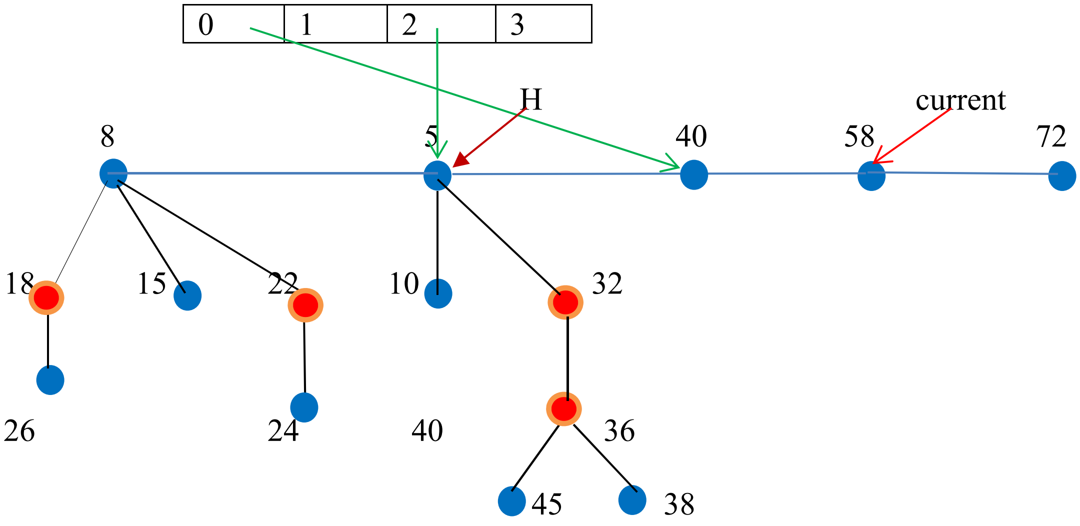
decreaseKey(q,-∞,H):

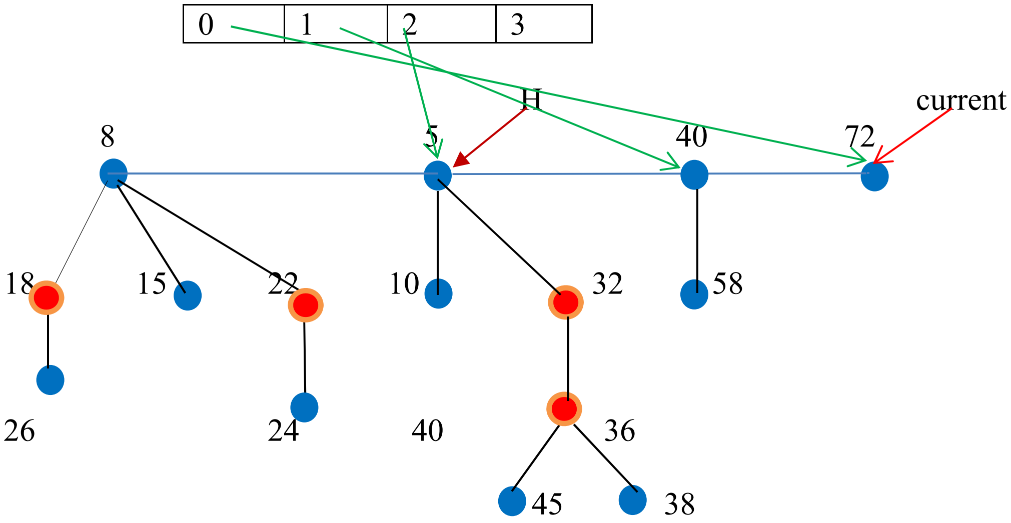


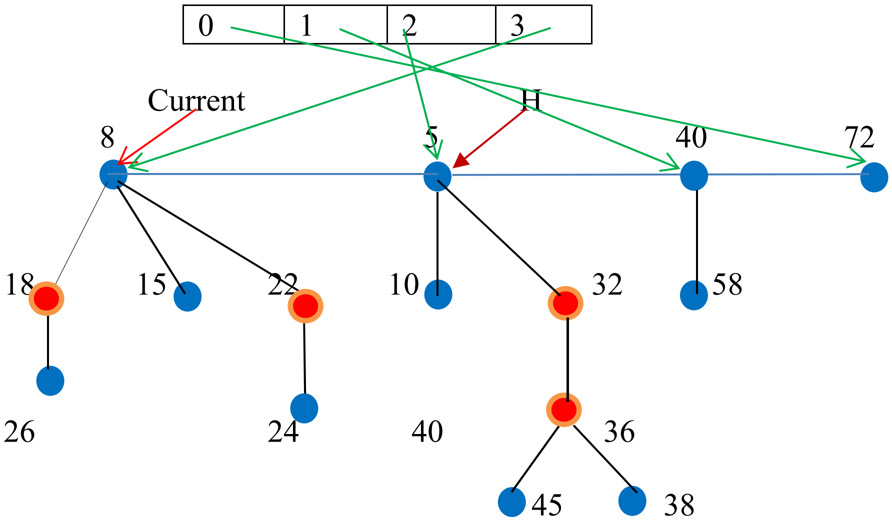
deleteMin(H):



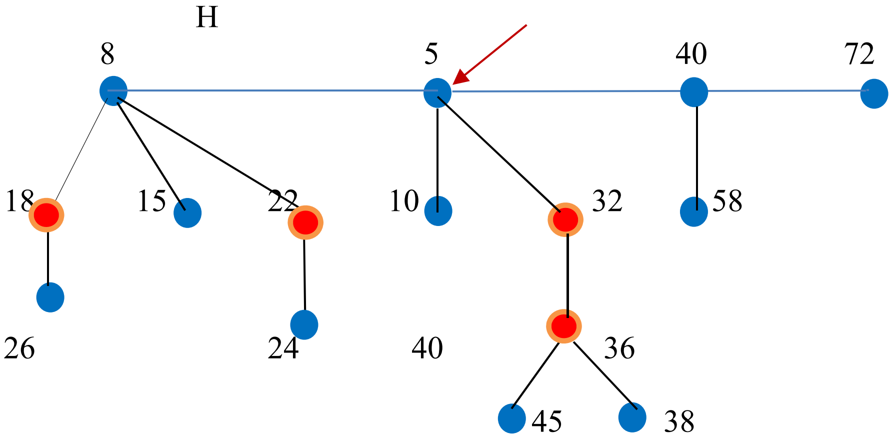
Tree consolidation:




0
1
2
3
H
current
8
5
40
72
18
15
22
10
32
58
26
24
40
36
45
38


0
1
2
3
Current
H
8
5
40
72
18
15
22
10
32
58
26
24
40
36
45
38

Final FH:


H
8
5
40
72
18
15
22
10
32
58
26
24
40
36
45
38

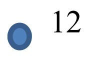
7. (20) Given a set of 8 records with priorities S = {12, 14, 16, 9, 7, 13, 10, 15}.
   (a) Construct an AVL tree for S by inserting the records, in the given order, into an initially empty AVL tree.
   (b) Construct an AVL tree for S by inserting the records, in the reversed given order, into an initially empty AVL tree.

   Remark: You must show your tree and indicate the type of rotation performed, if any, after each insert operation.

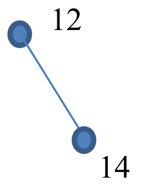
Solution:

(a) Insert in the given order:

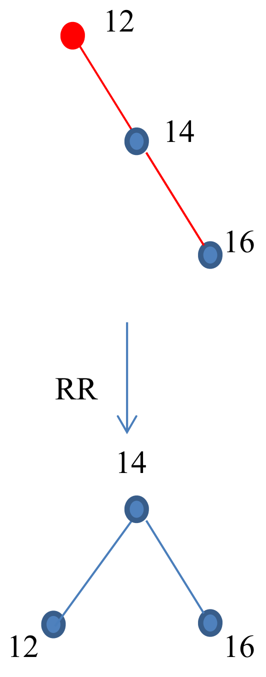
Insert(12):



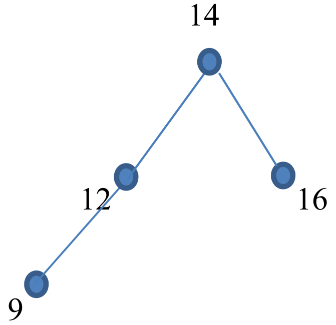
Insert(14):



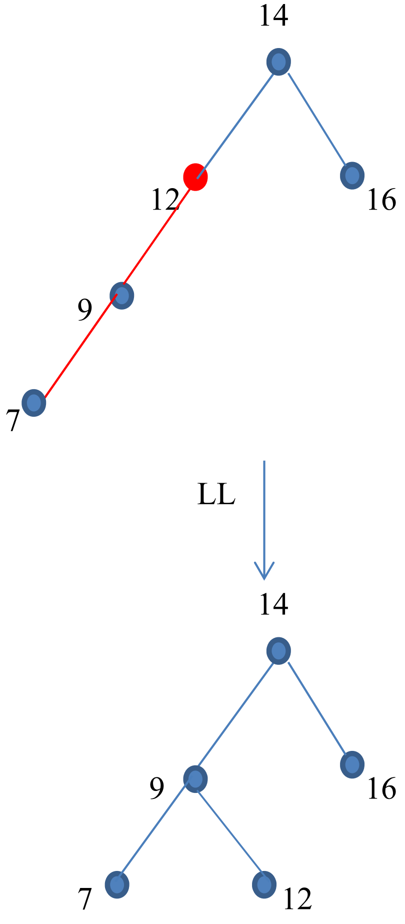
Insert(16):



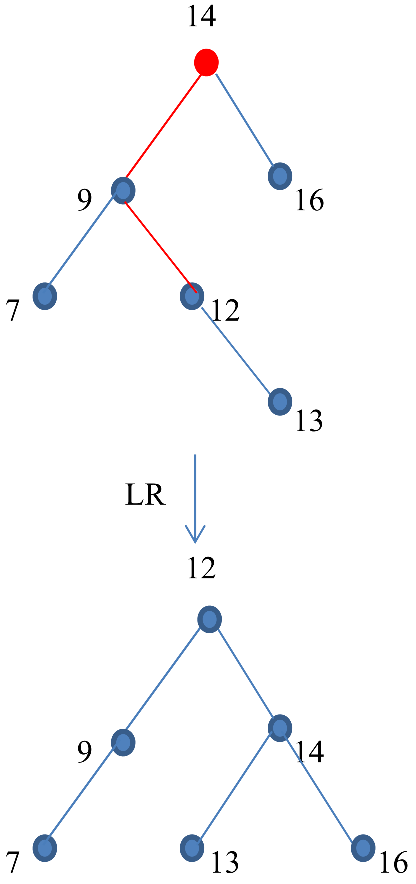
Insert(9):



Insert(7):

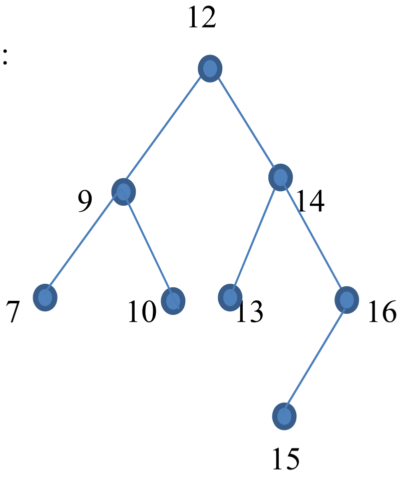


Insert(13):



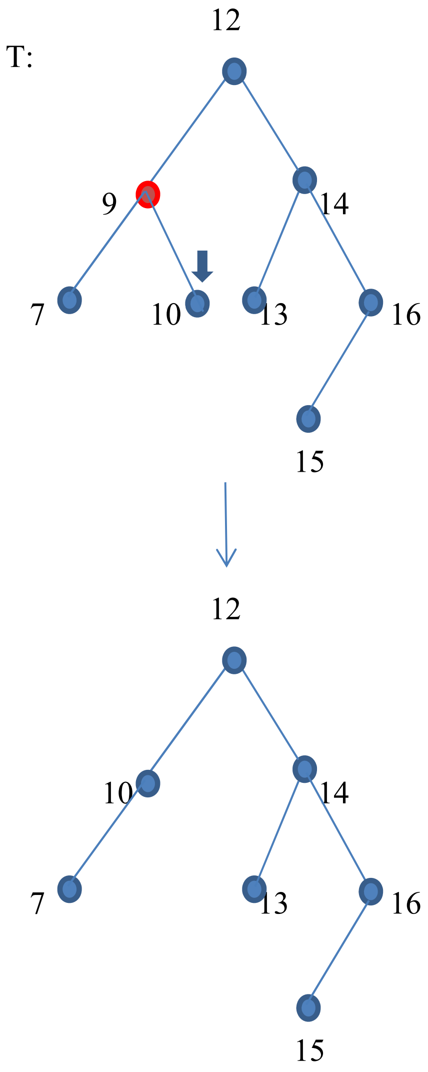
Insert(10) and Insert(15):

T:



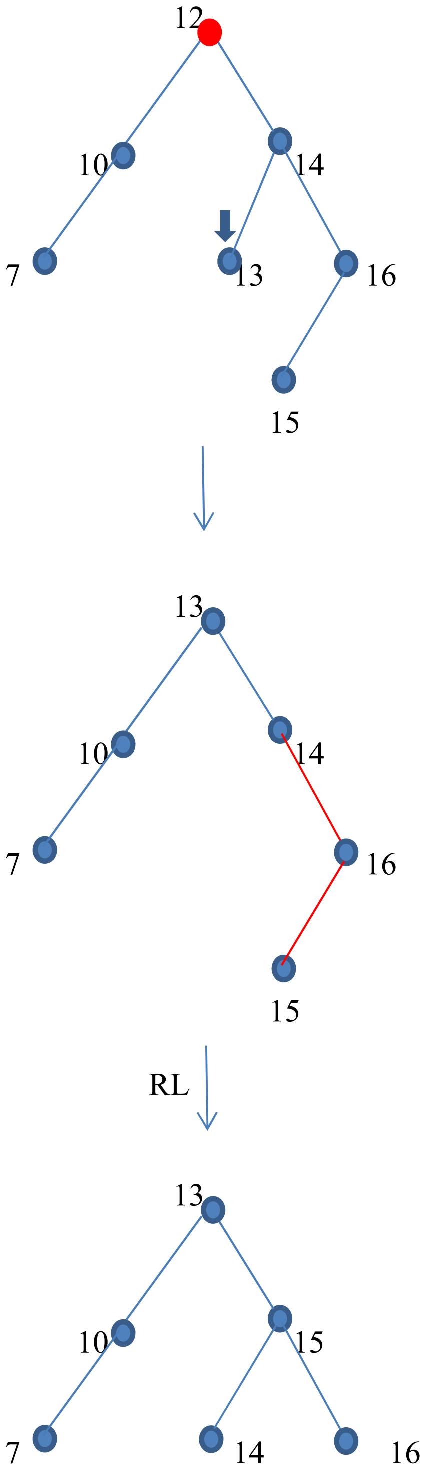
Examples of delete(x):

Delete(9):



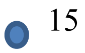
Remark: No rotation required.

Delete(12):

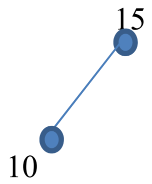


(b) Insert in the reversed given order:

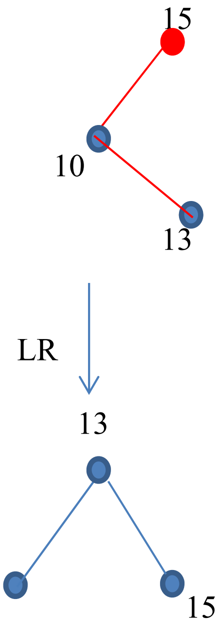
Insert(15):



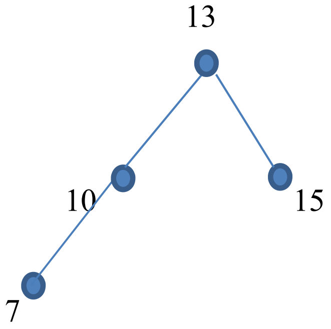
Insert(10):



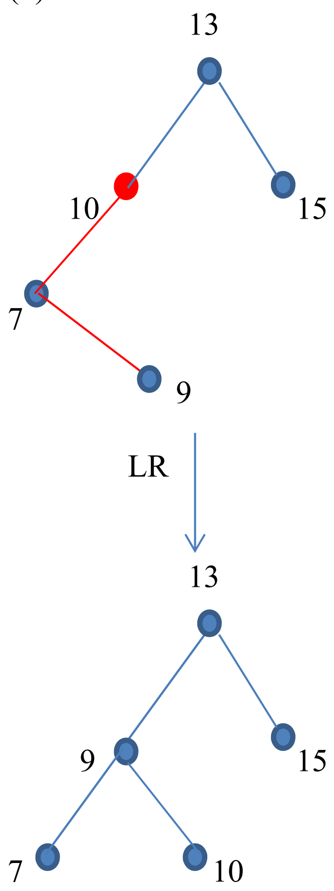
Insert(13):



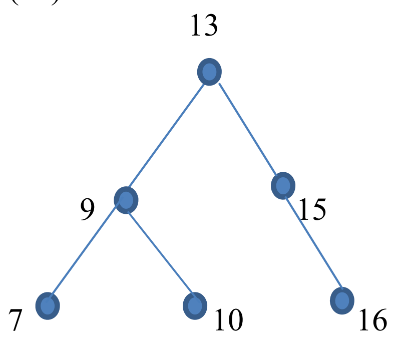
Insert(7):



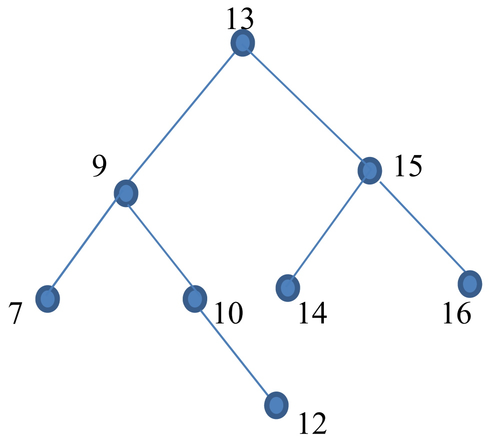
Insert(9):



Insert(16):



Insert(14) and Insert(12):



11/27/2018