1. Let H be a hash table of size 7 with the hash function $h(K) = 5 * k \bmod 7$ implemented using CLOSED (CHAINED) hashing. Consider the following sequence of insert operations:

`Insert(`6, D_6`);`
`Insert(`3, D_3`);`
`Insert(`4, D_4`);`
`Insert(`10, D_{10}`);`
`Insert(`20, D_{20}`);`
`Insert(`45, D_{45}`);`

Draw the hash table after all the above operations have been executed, showing which data elements are in which locations of the hash table and how they are stored. Show your work.

2. Let H be a hash table of size 7 with the hash function $h(K, j) = (5 * k + 3 * j) \bmod 7$ implemented using OPEN hashing.

Assume that $H[0], H[2], H[3]$ and $H[6]$ are already filled in the hash table so that the hash table looks like:

0	1	2	3	4	5	6
D_a		D_b	D_c			D_d

Consider the following sequence of insert operations:

`Insert(`2, D_e`);`
`Insert(`6, D_f`);`

Draw the hash table after all the above operations have been executed, showing which data elements are in which locations of the hash table. Show your work.

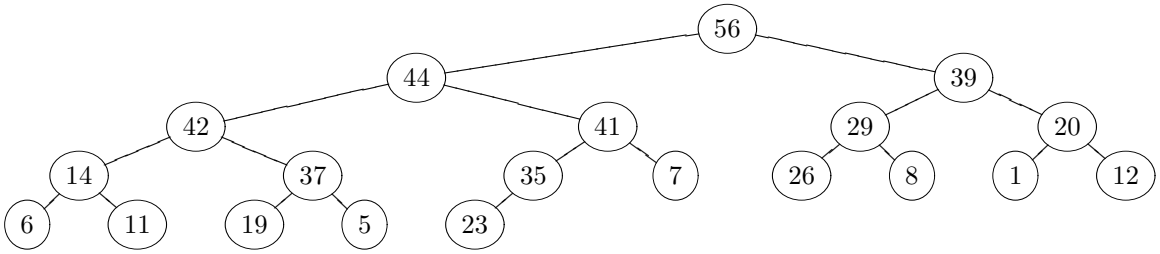
3. (a) Draw the binary tree produced by inserting the following elements in a max-heap in the given order:

$$4, 7, 2, 5, 3, 8, 9, 12$$

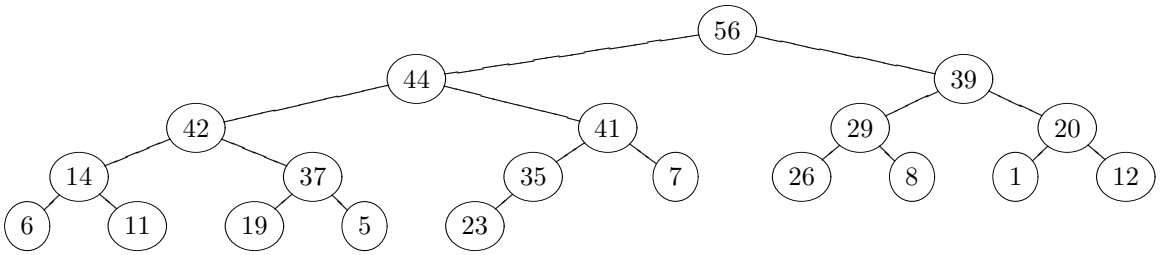
(Show your work.)

(b) List the elements of the array which represents the heap in problem 3a. List the elements in the order they appear in the array.

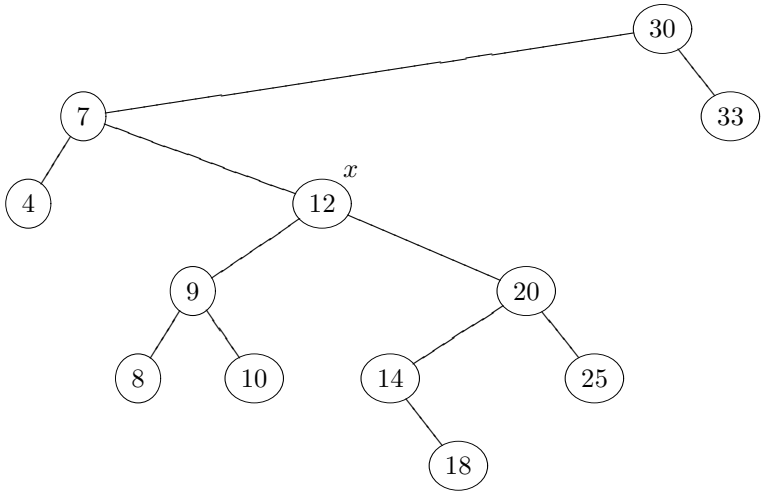
4. Apply the ExtractMax() operation to the following max heap and draw the resulting binary tree. (Show your work.)



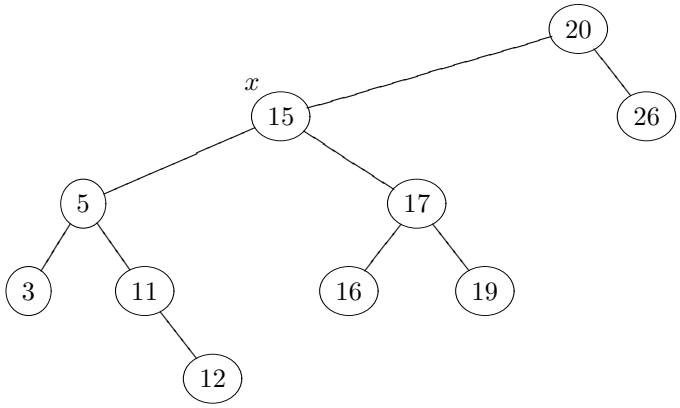
5. Apply the operation Insert(49) to the following max heap and draw the resulting binary tree. (Show your work.)



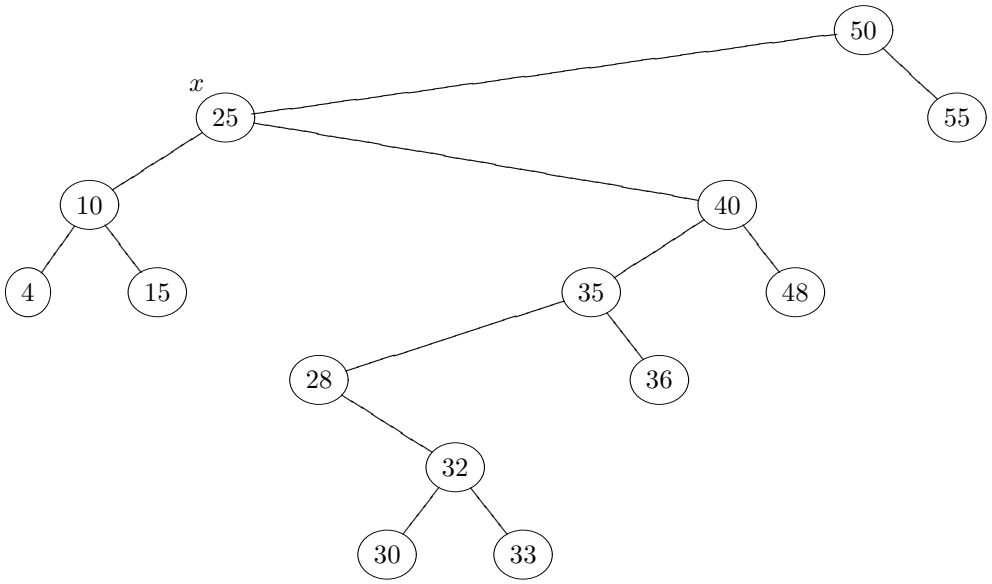
6. Draw the result of applying the LeftRotate operation (slide 5.38) to node x in the following binary search tree:



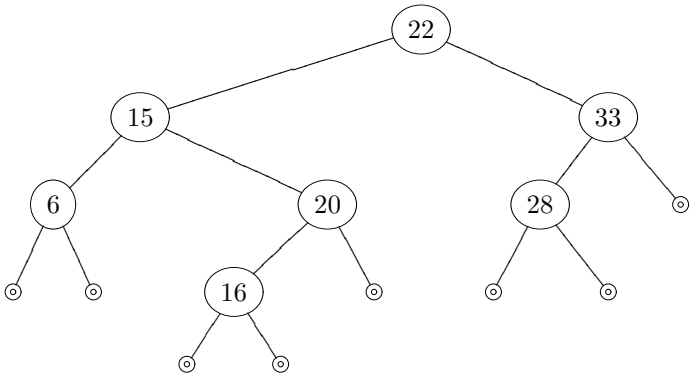
7. Draw the result of applying a right rotation to node x in the following binary search tree:



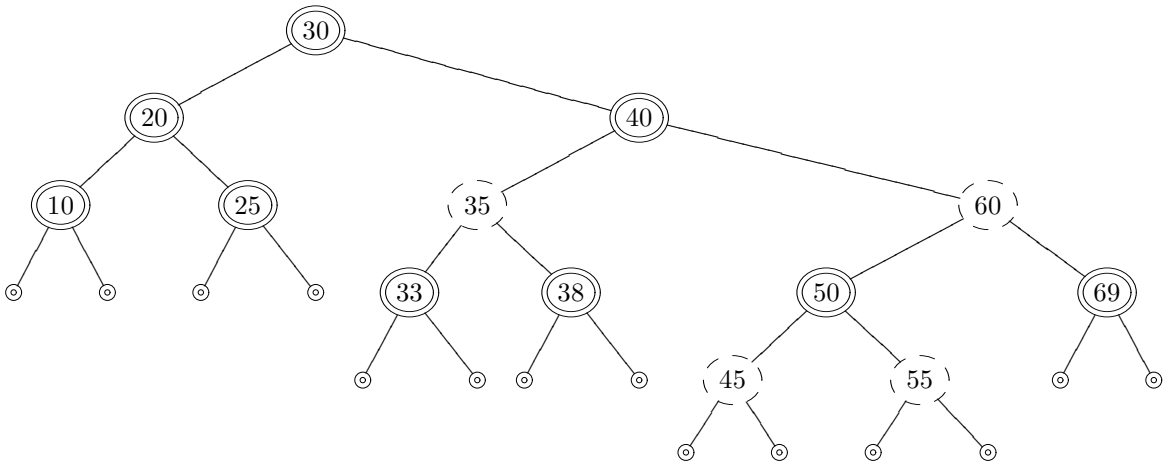
8. Draw the result of applying the TreeDelete operation (slide 5.25) to node x in the following binary search tree:



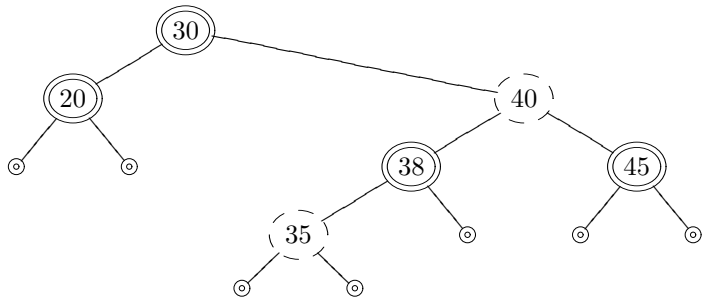
9. Color the nodes of the following binary search tree so that it is a red-black tree:



10. Draw the result of applying RBTreeInsert to insert 48 in the following red-black tree. (Black nodes have double circles. Red nodes have dashed circles.) Indicate which nodes are red or black in your drawing. Be sure that the resulting tree has all the properties of a red-black tree. Show your work, i.e., show the effects of the different algorithm steps on the tree.



11. Draw the result of applying RBTreeInsert to insert 31 in the following red-black tree. (Black nodes have double circles. Red nodes have dashed circles.) Indicate which nodes are red or black in your drawing. Be sure that the resulting tree has all the properties of a red-black tree. Show your work, i.e., show the effects of the different algorithm steps on the tree.



12. Draw the result of applying RBTreeInsert to insert 37 in the following red-black tree. (Black nodes have double circles. Red nodes have dashed circles.) Indicate which nodes are red or black in your drawing. Be sure that the resulting tree has all the properties of a red-black tree. Show your work, i.e., show the effects of the different algorithm steps on the tree.