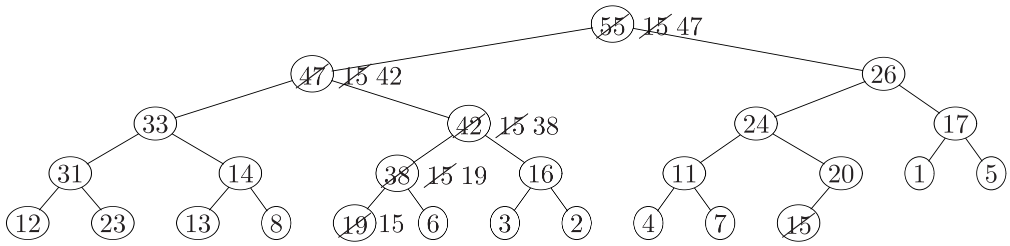
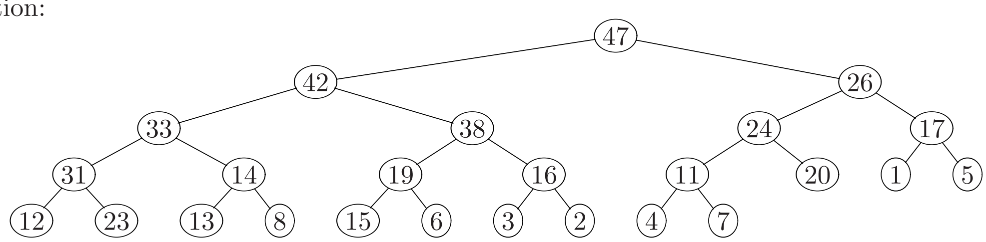
CSE 2331 Sample Midterm II Solutions

1. Apply ExtractMax() operation to the following heap:



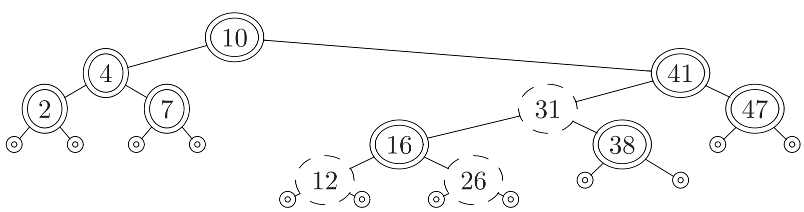
Solution:



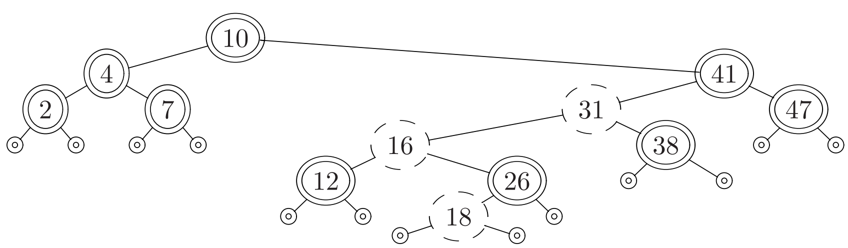
2. (a) Delete node z:
   - If z has no children, then delete z.
   - If z has one child, then replace z with its child.
   - If z has two children, then
     - Find the successor y of z (y is the minimum node in the subtree rooted at $z.right$.)
     - Replace y with its right child.
     - Replace z with y.

   (b) Worst case running time of delete node is $\Theta(h)$ where h is the height of the tree. The running time is dominated by the time to find the successor of z which could require going from the root to a leaf. All other operations take $\Theta(1)$ time.

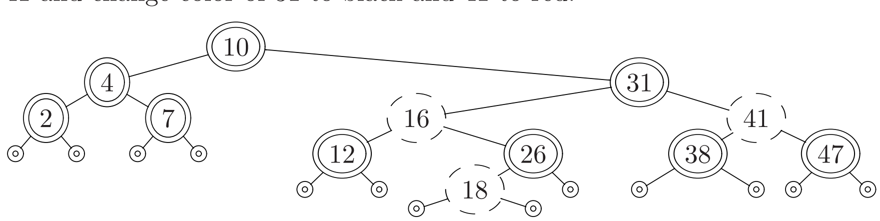
3. Apply RBTreeInsert to insert 18 in the following red-black tree.



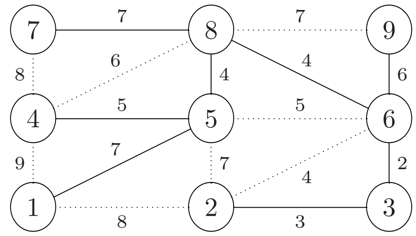
Insert 18 as a red left child of 26 and change color of 12 and 26 to black and of 16 to red:



Rotate on 41 and change color of 31 to black and 41 to red:



4. (a) MINIMUM SPANNING TREE:



(b) Edges in output order: $(1,5), (5,8), (8,6), (6,3), (3,2), (5,4), (6,9), (8,7)$.

5. Let P be a priority queue implemented by a data structure where P.Initialize() takes $\Theta(1)$ time, P.Insert() takes $\Theta(s^{1/3})$ time, and P.ExtractMax() takes $\Theta(s)$ time, where s is the number of elements in P. Analyze the following algorithm.

```
    Func1(A, n)
    /* A is an array of n elements                                  */
1   P.Initialize();
2   for i ← 1 to n do
3   |   for j ← 1 to 100 do
4   |   |   P.Insert(A[i] * A[j]);
5   |   end
6   end
7   s ← 0;
8   for i ← 1 to n do
9   |   s ← s+ P.ExtractMax();
10  end
11  return (s);
```

Steps 2 and 3 execute a total of $100n$ times, so there are $100n$ insert operations.
Cost of insert:

$$\sum_{s=1}^{100n} s^{1/3} \leq \sum_{s=1}^{100n} (100n)^{1/3} = (100n)^{4/3}.$$

$$\begin{aligned}\sum_{s=1}^{100n} s^{1/3} &\geq \sum_{s=100n/2}^{100n} s^{1/3} \geq \sum_{s=100n/2}^{100n} ((100n)/2)^{1/3} \\ &= \sum_{s=100n/2}^{100n} (n)^{1/3} 50^{1/3} = n^{4/3}(50)^{4/3}.\end{aligned}$$

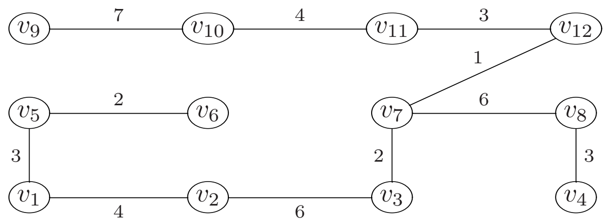
Cost of extract max:

$$\sum_{i=1}^{n} (100n - i) \leq \sum_{i=1}^{n} 100n = 100n^2.$$

$$\sum_{i=1}^{n} (100n - i) \geq \sum_{i=1}^{n} (100n - n) = 99n^2.$$

Total cost is $\Theta(n^{4/3} + n^2) = \Theta(n^2)$.

6. Let G be an edge weighted graph whose MINIMUM SPANNING TREE is:

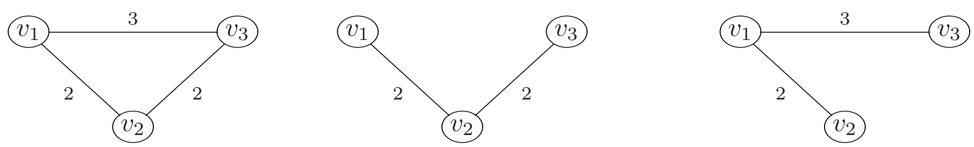


Assume that the graph G contains the edge (v_6, v_{10}).
Prove that the weight of (v_6, v_{10}) is NOT 5.

Proof. If edge (v_6, v_{10}) had weight 5, then we could replace edge (v_2, v_3) by edge (v_6, v_{10}) and reduce the cost of the spanning tree by $6 - 5 = 1$. Thus, the given tree would not be a minimum spanning tree of graph G. We conclude that edge (v_6, v_{10}) must not have weight 5 in G. $\square$

7. Give an example of an edge weighted graph with at most 6 edges whose minimum spanning tree is not the same as the shortest path tree from vertex v_1.

One possible solution:



(a) Graph. (b) Min span tree. (c) Shortest path tree from v_1.

8. Let G be an edge weighted graph with 100 vertices where the distance of the shortest path from v_1 to v_{99} is 500 and the distance of the shortest path from v_1 to v_{100} is 510.

If G has an edge (v_{99}, v_{100}), then weight(v_{99}, v_{100}) does not equal 7.

Proof. Assume G had an edge (v_{99}, v_{100}) with weight 7.
Let P be the shortest path from v_1 to v_{99}.
$P \cup (v_{99}, v_{100})$ has length $500 + 7 = 507 < 510$, contradicting the assumption that the shortest path from v_1 to v_{100} has length 510.
Therefore, G cannot have an edge (v_{99}, v_{100}) with weight 7. $\square$