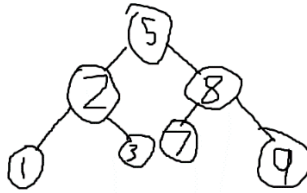


1. The root is 2. The leaves are 1, 4, and 0. Every other node is an internal node
2. Pre: 2,7,3,1,5,9,4,0
In-Order: 1,3,7,2,4,9,5,0
Post: 0,4,9,5,1,3,7,2
3. A, B, C, D
4. 1,3,4,6,7,8,10,13,14



- 5.
6. 3 and 7

```

47
48
49
50
51
52
53
54
55
56
57
58 /**
59  * Gets the breadth-first search traversal of the tree with "this" as root
60  * Each node's data should have the format (data) Eg: (5) and (10)
61  * @return the bfs traversal
62  */
63
64 public static String breadthFirstSearch(BinaryNode root) {
65     Queue<BinaryNode> queue = new LinkedList<BinaryNode>();
66     String bfs = "";
67     //put root on queue to start it off
68     queue.add(root);
69     while(!queue.isEmpty()) {
70         bfs += "(" + queue.peek().data + ")";
71         if(queue.peek().leftChild != null) {
72             queue.add(queue.peek().leftChild);
73         }
74         if(queue.peek().rightChild != null) {
75             queue.add(queue.peek().rightChild);
76         }
77         queue.remove(queue.peek());
78     }
79     return bfs;
80 }
81

```

Console Output:

```

-terminated: BinaryNodeTest [JUnit] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 19, 2020, 10:29:18 PM)
(2)(7)(5)(3)(9)(0)(1)(4)
(8)(3)(10)(1)(6)(14)(4)(7)(13)

```