CSCI-2725 Homework 3

Yitao Tian

March 17, 2023

Extra credit: There is 5 percentage extra credit if you don't submit hand-written homework (including the diagrams). You can use LaTeX or any other tool to write your homework.

1. (4 points each) **State** the following as true or false.

   (a) Recursive functions often have fewer local variables than the equivalent iterative functions.

   Answer: True; though more intermediate parameters.

   (b) Recursive functions must contain a path that does not contain recursive call.

   Answer: True; this path is called the "base case".

   (c) Recursive functions are always less efficient than the corresponding iterative functions in terms of Big-O complexity.

   Answer: False; not always, but typically.

2. (6 points each)

```
int fun(int num) {
    if (num < 2) {
        return 1;
    }
    else if (num % 2 == 0) {
        return fun(num / 2);
    }
    else {
        return fun(3 * num + 1);
    }
}
```

(a) How many recursive calls will be made for function call `fun(7)`? Show your work on how you got the number of recursive calls.

Answer: Seventeen (17) function calls including first call (`fun(7)`).
Justification: fun(7), fun(22), fun(11), fun(34), fun(17), fun(52), fun(26), fun(13), fun(40), fun(20), fun(10), fun(5), fun(16), fun(8), fun(4), fun(2), fun(1)

(b) How many recursive calls will be made for function call `fun(8)`? Show your work on how you got the number of recursive call.

Answer: Four (4) function calls including first call (`fun(8)`).
Justification: fun(8), fun(4), fun(2), fun(1)

(c) Do you see any problem in the above function? If so, give justification to your answer.

Answer: Yes, there is a problem; the function should not have wildly different complexity for its inputs.

3. (6 points each) What are the outputs of the following function calls? Show your work to justify your answer.

```
int fun(int k, int n)
{
    if (n == k) { return k; }
    else {
        if (n > k) { return fun(k, n - k); }
        else { return fun(k - n, n); }
    }
}
```

(a) `System.out.print(fun(10,6));`

Answer: prints 2

Justification: fun(10,6), fun(4,6), fun(4,2), fun(2,2)

(b) `System.out.print(fun(7,13));`

Answer: prints 1

Justification: fun(7,13), fun(7,6), fun(1,6), fun(1,5), fun(1,4), fun(1,3), fun(1,2), fun(1,1)

4. (6 points) What is the output of the following function call? Show your work to justify your answer.

```
1       void Quiz( int n ) {
2           if (n > 0) {
3               System.out.print("0");
4               Quiz(n - 1);
5               System.out.print("1");
6               Quiz(n - 1);
7           }
8       }
```

~~`System.out.println(Quiz(2));`~~ `// not valid Java code`
`Quiz(2);`

Answer: 0, 0, 1, 1, 0, 1

Justification: Quiz(2), print 0, Quiz(1), print 0, Quiz(0), print 1, Quiz(0), print 1, Quiz(1), print 0, Quiz(0), print 1, Quiz(0)

5. (4 points each) **Consider** the binary search tree below (the numbers inside the nodes are not the actual data; they are just numbers labeling the nodes).

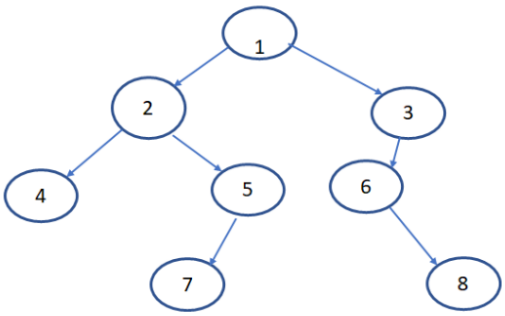

Figure 1: A binary search tree

(a) If node 1 is deleted, what node or nodes can replace node 1 in the tree?

Answer: Nodes 5 and 6 can replace; they are the predecessor and successor, respectively.

(b) 4 2 7 5 1 6 8 3 is a traversal of the tree in which order?

Answer: In "in-order" order.

(c) 1 2 4 5 7 3 6 8 is a traversal of the tree in which order?

Answer: In "pre-order" order.

6. (4 points each) **Consider** the following set of input data:

50 72 96 94 100 26 13 11 9 2 10 14 25 51 33 16 95 12

(a) **Draw** a binary search tree using the above input (in the order given).

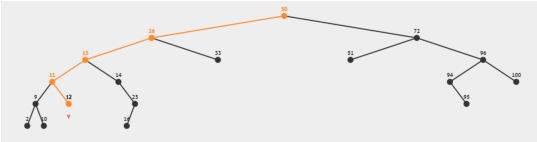

Figure 2: A BST based on the given, ordered input

(b) What nodes are in level 3 of the tree?

Answer: 11, 14, 94, 100 (assuming root is level 0).

(c) **List** all the leaf nodes in the tree.

Answer: 2, 10, 12, 16, 33, 51, 95, 100.

(d) What is the height of the binary search tree containing this input?

Answer: 5 excluding the root, 6 including.

(e) **Give** the output of post-order traversal function called on the above tree.

Answer: 2 10 9 12 11 16 25 14 13 33 26 51 95 94 100 96 72 50

(f) What is the maximum height of the binary search tree containing this input?

Answer: 17 excluding the root, 18 including.

(g) What is the minimum height of a binary search tree containing this input?

Answer: 4 excluding the root, 5 including.

(h) **Draw** the tree after deleting the node containing 94.

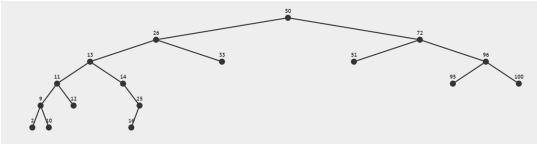

Figure 3: The BST after deleting 94

(i) **Use** the above tree (from part (h)) and draw the tree after deleting the node containing 13.

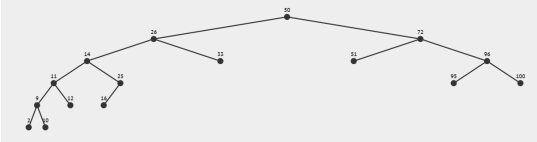

Figure 4: The BST after deleting 13

(j) **Use** the above tree (from part (i)) and draw the tree after deleting the node containing 50.

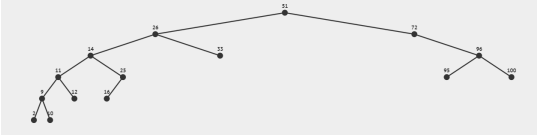

Figure 5: The BST after deleting 50