Name: Daniel Detore Date: 10/31/2024

I pledge my honor that I have abided by the Stevens Honor System.

Point values are assigned for each question. Points earned: \_\_\_\_ / 100

1. Consider the algorithm on page 148 in the textbook for Binary Reflected Gray Codes. What change(s) would you make so that it generates the usual binary numbers **in order** for a given length n? Your algorithm must be recursive and keep the same structure as the one in the textbook. Describe only the change(s). (10 points)

**Copy list L1 to L2 in order, rather than reverse order.**

1. Show the steps to multiply 72 x 93 with Russian peasant multiplication, as seen in Figure 4.11b on page 154 in the textbook. (10 points)

72 \* 93

36 \* 186

18 \* 372

9 \* 744

4 \* 1488

2 \* 2976

1 \* 5952

72 \* 93 = 5952 + 744 = **6696**

1. Suppose you use the LomutoPartition() function on page 159 in the textbook in your implementation of Quicksort. (10 points, 5 points each)
   1. Describe the types of input that cause Quicksort to perform its worst-case running time. Explain why these types of input cause the worst-case.

**In the worst case, the array is already sorted either forward or backwards. Since this implementation of Lomuto partitioning uses the first element as the pivot, it will never have a balanced partition, meaning it cannot divide and conquer, therefore it will take the longest possible time.**

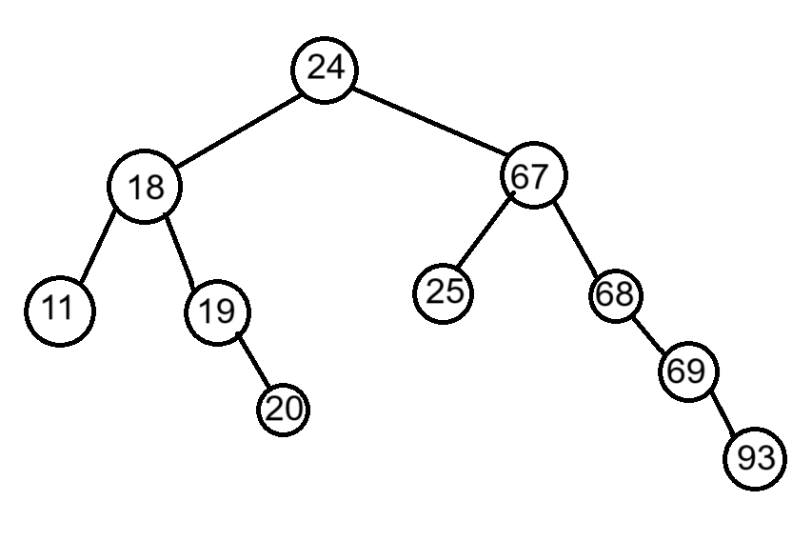
* 1. What is that worst-case running time?

1. Compute 2205 x 1132 by applying the divide-and-conquer Karatsuba algorithm outlined in the text. Repeat the process until the numbers being multiplied are each 1 digit. For each multiplication, show the values of c2, c1, and c0. Do not skip steps. (10 points)

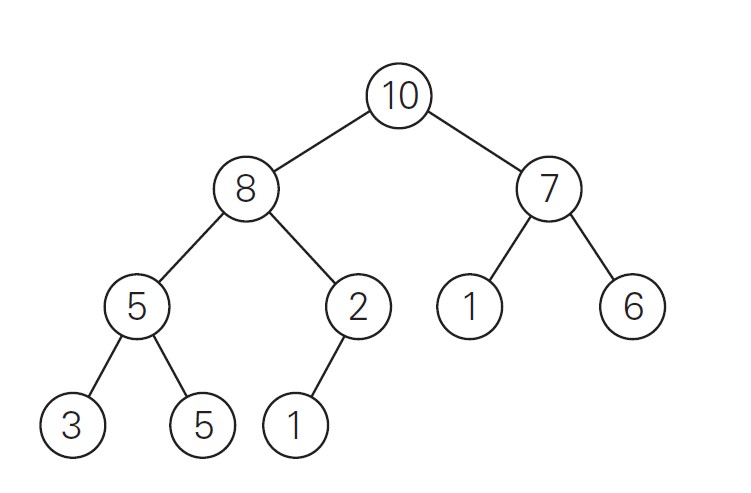
A white paper with black text

Description automatically generated

1. Draw the binary search tree after inserting the following keys: 24 18 67 68 69 25 19 20 11 93   
   (10 points)



1. Consider the following binary tree. (16 points, 2 points each)



* 1. Traverse the tree preorder.

**10, 8, 5, 3, 5, 2, 1, 7, 1, 6**

* 1. Traverse the tree inorder.

**3, 5, 5, 8, 1, 2, 10, 1, 7, 6**

* 1. Traverse the tree postorder.

**3, 5, 5, 1, 2, 8, 1, 6, 7, 10**

* 1. How many internal nodes are there? **5**
  2. How many leaves are there? **5**
  3. What is the maximum width of the tree? **4**
  4. What is the height of the tree? **4**
  5. What is the diameter of the tree? **5**

1. Use the Master Theorem to give tight asymptotic bounds for the following recurrences. (25 points, 5 points each)
   1. 𝑇(𝑛) = 2𝑇(𝑛/4) + 1
   2. 𝑇(𝑛) = 2𝑇(𝑛/4) +
   3. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛
   4. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛2
   5. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛3
2. Consider the following function. (9 points)

int function(int n) {

if(n <= 1) {

return 0;

}

int temp = 0;

for(int i = 1; i <= 6; i++) {

temp += function(n / 3);

}

for(int i = 1; i <= n; i++) {

for(int j = 1; j \* j <= n; j++) {

temp++;

}

}

return temp;

}

1. Write an expression for the runtime 𝑇(𝑛) for the function (with the correct asymptotic symbol for the *f(n)* part of the relation). (4 points)
2. Use the Master Theorem to give a tight asymptotic bound. Simplify your answer as much as possible. (5 points)