Name: Breona Pizzuta Date: 10/28/24

“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)

A white text with black text

Description automatically generated

To generate the usual binary numbers in order I would copy list L1 to list L2 in order rather than in reversed 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)

|  |  |  |  |
| --- | --- | --- | --- |
| Step1:  N= 72 m= 93 | Step 2:  N m  72 93  36 186  18 372  9 744  4 1488  2 2976   1. 5952 | Step 3:  N m  72  36  18  9 744  4   1. 5952 | Step 4: 744+ 5952 = 6696= 9\*744 |

Answer: 6696

1. Suppose you use the LomutoPartition() function on page 159 in the textbook in your implementation of Quicksort. (10 points, 5 points each)

A diagram of a diagram

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated

* 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 each partition results in a very unbalanced partitioning of the array where one segment has size n - 1 and the other segment has size zero, this happens if the array is already sorted in non-decreasing order or non-increasing order.

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

The worst case running time is Θ(n²)

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 background with black text

Description automatically generated

2205 and 1132

A1= 22 and A0= 05 B1= 11 and B0= 32

C0= 5\*32= 160

A=05= 0\*10 +5

A1=0 A0= 5

B= 32= 3\*10 +2

B1= 3, B0=2

C0= 5\*2= 10

C1= 5\*5 =25

C2=0\*3= 0

A\*B= 0\*100+(25-0-10)\*10+10 = 150+10= 160

C1= (22+5) \* (11+32) = 27\*43= 1161

A= 27 = 2\*10+7

A1= 2, A0= 7

B= 43= 4\*10+3

B1= 4, B0=3

C0= 7\*3= 21

C1= 9\*7= 63

C2= 2\*4= 8

A\*B= 8\*100 + (63-8-21) \*10 +21= 1161

C2= 22\*11 = 242

A= 22= 2\*10+2

A1= 2, A0= 2

B= 1\*10+1

B1=1, B0=1

C0= 2\*1= 2

C1= 4\*2 = 8

C2= 2\*1= 2

A\*B= 2\*100 + (8-2-2) \*10 +2= 242

A\*B= 242\*10000 + (1161-242-160)\*100+160

=242\*100000 + 759\*100+160

= 2420000+ 75900+160

= 2496060

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

24

/ \

18 67

/ \ / \

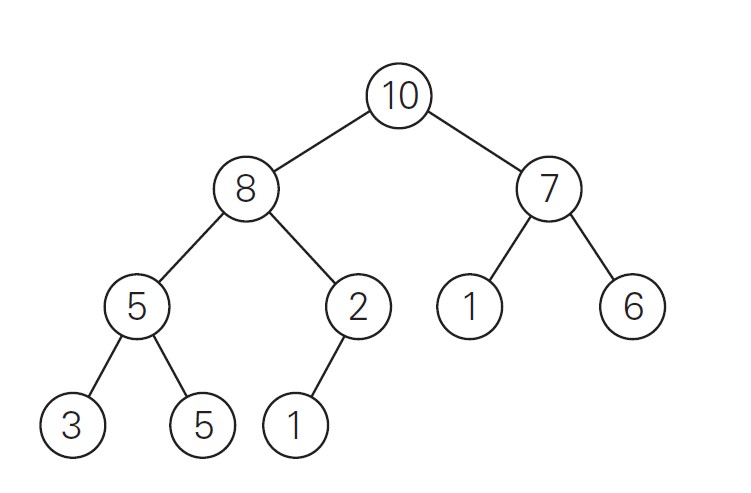
11 19 25 68

\ \

20 69 \

93

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 internal nodes

* 1. How many leaves are there?

5 leaves

* 1. What is the maximum width of the tree?

4 is the maximum width of the tree

* 1. What is the height of the tree?

3 is the height of the tree

* 1. What is the diameter of the tree?

5 is the diameter of the tree

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

A= 2, B=4, d= 0

* 1. 𝑇(𝑛) = 2𝑇(𝑛/4) +

A= 2, B=4, d= 1/2

* 1. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛

A= 2, B=4, d= 1

* 1. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛2

A= 2, B=4, d= 2

* 1. 𝑇(𝑛) = 2𝑇(𝑛/4) + 𝑛3

A= 2, B=4, d= 3

1. 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)

T(n)= 6T(n/3) + (n

1. Use the Master Theorem to give a tight asymptotic bound. Simplify your answer as much as possible. (5 points)

A= 6, B=3, d= 3/2