CS 385, Homework 4: Decrease/Divide and Conquer
"I pledge my honor that I have abided by the Stevens Honor System."

Name: Julia Nelson Date: March 31, 2020

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 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)

> To generate binary numbers *in order* for a given length n while painting the recursive structure, I would change the line
> —> copy list L1 to l2 in reverse order
> to
> —> copy list L1 to L2 in the same order

2. 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)

Russian Peasant 72 x 93

n	*m*	
72	93	
36	186	
18	372	
9	744 —	744
4	1488	
2	2976	
1	5952 —	5952
		= 6696

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

> Worst-case running time of quicksort caused by an input that is sorted in Non-decreasing or Non-increasing order (Lomuto partitions size 1 or n-1)

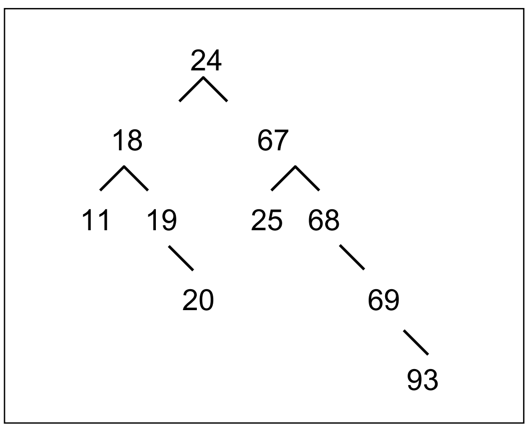
   b. What is that running time?

> T(1) = *theta*(1)
> T(n) = T(n-1) + *theta*(n)
> $= E^{n}_{k=1}$ (*theta*(k)) = *theta*(n^2)

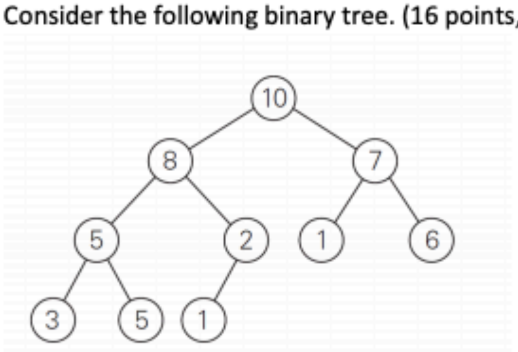
4. Compute 2205 x 1132 by applying the divide-and-conquer algorithm outlined in the text. Repeat the process until the numbers being multiplied are each 1 digit. For each multiplication, show the values of c_2, c_1, and c_0. Do not skip steps. (10 points)

a = 2205 b = 1132
c_2 = 22 * 11
  c_2 = 2 * 1 = 2
  c_0 = 2 * 1 = 2
  c_1 = (2 + 1) *(2+1) - (2+2) = 3*3-4 = 5
  C = 200+ 50 + 2 =252
c_0 = 05 * 32
  c_2 = 0 * 3 = 0
  c_0 = 5 * 2 = 10
  c_1 = (0+5) *(3+2) - (0+10) = 5*5 -10 = 15
  C = 0 + 150 + 10 = 160
c_1 = (22 + 05) * (11 + 32) - (c_2 + c_0)
……….. final answer = 2496060

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



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



a) Traverse the tree preorder.
b) Traverse the tree inorder.
c) Traverse the tree postorder.
d) How many internal nodes are there?
e) How many leaves are there?
f) What is the maximum width of the tree?
g) What is the height of the tree?
h) What is the diameter of the tree?

a — 10, 8, 5, 3, 5, 2, 1, 7, 1, 6
b — 3, 5, 5, 8, 1, 2, 10, 1, 7, 6
c — 3, 5, 5, 1, 2, 8, 1, 6, 7, 10
d — 5, 2, 8, 7, 10 = 5
e — 3, 5, 1, 1, 6 = 5
f — max width = 4
g — height = 3
h — diameter = 5

7. Use the Master Theorem to give tight asymptotic bounds for the following recurrences. (25 points, 5 points each)

a) $T(n) = 2T(n/4) + 1$
b) $T(n) = 2T(n/4) + \sqrt{n}$
c) $T(n) = 2T(n/4) + n$
d) $T(n) = 2T(n/4) + n^2$
e) $T(n) = 2T(n/4) + n^3$

a) T(n) = 2T(n/4) + 1 a = 2 a >= 1 b = 4 b >= 1 d = 0 d >= 0 $2 > 4^0$ T(n) e *theta*($n^{\log_4 2}$) **= T(n) e *theta*(sqrt(n))**	b) T(n) = 2T(n/4) + sqrt(n) a = 2 b = 4 d = 1/2 $2 >= 4^{1/2}$ 2 = 2 **= T(n) e *theta*(sqrt(n)*log4(n))**

c) T(n) = 2T(n/4) + n a = 2 b = 4 d = 1 $2 <= 4^1$ **= T(n) e *theta*(n)**	d) T(n) = 2T(n/4) + n^2 a = 2 b = 4 d = 2 $2 < 4^2$ **= T(n) e *theta*(n^2)**	e) T(n) = 2T(n/4) + n^3 a = 2 b = 4 d = 3 $2 < 4^3$ **= T(n) e *theta*(n^3)**

8. 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;
}
```

a) Write an expression for the runtime $T(n)$ for the function. (4 points)
b) Use the Master Theorem to give a tight asymptotic bound. Simplify your answer as much as possible. (5 points)

a) T(n) = *theta*(1) + *theta*(1) + 6T(n/3) + *theta*(n * sqrt(n))
T(n) = 6T(n/3) + *theta*(n * sqrt(n))
T(n) = 6T(n/3) + $n^{3/2}$

b) T(n) = 6T(n/3) + $n^{3/2}$
a = 6
b = 3
d = 3/2 6 > 5.196
T(n) e *theta*($n^{\log_3 6}$)
T(n) e *theta*($n^{1.6309}$)