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)

**ALGORITHM** *BRGC*($n$)
//Generates recursively the binary reflected Gray code of order $n$
//Input: A positive integer $n$
//Output: A list of all bit strings of length $n$ composing the Gray code
**if** $n = 1$ make list $L$ containing bit strings 0 and 1 in this order
**else** generate list $L1$ of bit strings of size $n - 1$ by calling $BRGC(n - 1)$
copy list $L1$ to list $L2$ in reversed order
add 0 in front of each bit string in list $L1$
add 1 in front of each bit string in list $L2$
append $L2$ to $L1$ to get list $L$
**return** $L$

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

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

Answer: 6696

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

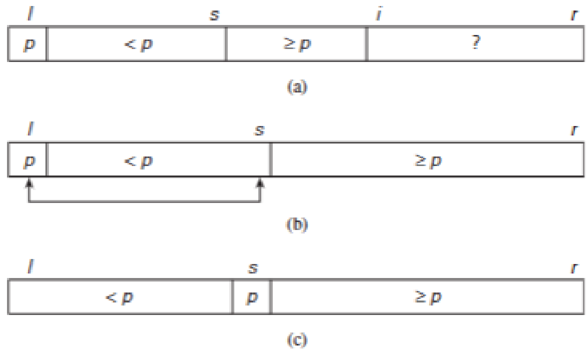



**FIGURE 4.13** Illustration of the Lomuto partitioning.

element in the first segment, swapping $A[i]$ and $A[s]$, and then incrementing $i$ to point to the new first element of the shrunk unprocessed segment. After no unprocessed elements remain (Figure 4.13b), the algorithm swaps the pivot with $A[s]$ to achieve a partition being sought (Figure 4.13c).

Here is pseudocode implementing this partitioning procedure.

**ALGORITHM** *LomutoPartition*$(A[l..r])$

```
//Partitions subarray by Lomuto's algorithm using first element as pivot
//Input: A subarray A[l..r] of array A[0..n − 1], defined by its left and right
//       indices l and r (l ≤ r)
//Output: Partition of A[l..r] and the new position of the pivot
p ← A[l]
s ← l
for i ← l + 1 to r do
    if A[i] < p
        s ← s + 1;  swap(A[s], A[i])
swap(A[l], A[s])
return s
```

How can we take advantage of a list partition to find the $k$th smallest element in it? Let us assume that the list is implemented as an array whose elements are indexed starting with a 0, and let $s$ be the partition's split position, i.e., the index of the array's element occupied by the pivot after partitioning. If $s = k - 1$, pivot $p$ itself is obviously the $k$th smallest element, which solves the problem. If $s > k - 1$, the $k$th smallest element in the entire array can be found as the $k$th smallest element in the left part of the partitioned array. And if $s < k - 1$, it can

a) 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.

b) What is that worst-case running time?

The worst case running time is $\Theta(n^2)$

4. 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 $c_2$, $c_1$, and $c_0$. Do not skip steps. (10 points)

So below is the definition of the Karatsuba Algorithm. Let's call:

$c_0 = a_0 * b_0$
$c_1 = (a_1 + a_0) * (b_1 + b_0)$
$c_2 = a_1 * b_1$
Then we have:
$a * b = c_2 * B^n + (c_1 - c_2 - c_0) * B^{(n/2)} + c_0$

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

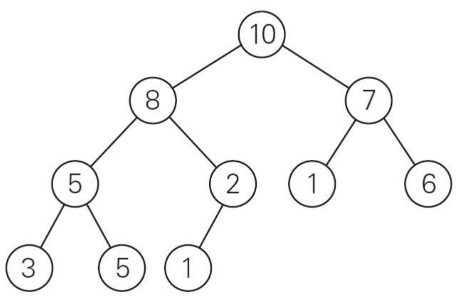
A*B= 242*10000 + (1161-242-160)*100+160

=242*100000 + 759*100+160
= 2420000+ 75900+160
= 2496060

5. 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
```

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




a) Traverse the tree preorder.
10,8,5,3,5,2,1,7,1,6

b) Traverse the tree inorder.
3, 5, 5, 8, 1, 2, 10, 1, 7, 6

c) Traverse the tree postorder.
3, 5, 5, 1, 2, 8, 1, 6, 7, 10

d) How many internal nodes are there?
5 internal nodes

e) How many leaves are there?
5 leaves

f) What is the maximum width of the tree?
4 is the maximum width of the tree

g) What is the height of the tree?

3 is the height of the tree

h) What is the diameter of the tree?

5 is the diameter of the tree

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

a) $T(n) = 2T(n/4) + 1$

A= 2, B=4, d= 0

$a > b^d \quad so\ \theta\left(n^{log_4 2}\right) = \theta\left(n^{1/2}\right)$

b) $T(n) = 2T(n/4) + \sqrt{n}$

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

$a = b^d \quad so\ \theta(\sqrt{n}\ log_4 n)$

c) $T(n) = 2T(n/4) + n$

A= 2, B=4, d= 1

$a < b^d \quad so\ \theta(n^1)$

d) $T(n) = 2T(n/4) + n^2$

A= 2, B=4, d= 2

$a < b^d \quad so\ \theta(n^2)$

e) $T(n) = 2T(n/4) + n^3$

A= 2, B=4, d= 3

$a < b^d \quad so\ \theta(n^3)$

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 (with the correct asymptotic symbol for the *f(n)* part of the relation). (4 points)

T(n)= 6T(n/3) + $\theta$ (n$\sqrt{n}$)

b) 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
$a > b^d \quad so\ \theta(n^{log_3 6})$