CS 206 A: 2021 SPRING QUIZ

Worth 200 points

- **The total score in this exam paper is 100 points.**
- **100 points will be converted to 200 points by multiplying your earned score by 2.**

QUESTION 0. INFORMATION (WORTH 2 POINTS)

Write your name and KAIST ID number at the top of **EACH** page on your answer sheet.

QUESTION 1 (13.5 POINTS)

1) Draw a full binary tree T that has the preorder and inorder as follows. Each node stores a single letter. (9 points)

Preorder	C T U W X S A Z O
Inorder	W U X T S C Z A O

2) What is the postorder of the full binary tree T? (4.5 points)

QUESTION 2 (10 POINTS)

Consider the following binary heap—the array a[] representation of a heap-ordered complete binary tree—which would represent a max heap, except that **ONE** integer key does not satisfy the max-heap property. (Note: As we did in class, indices start at one.)

i	0	1	2	3	4	5	6	7	8	9
a[i]	-	10	8	5	7	2	6	4	3	1

1) Draw the binary heap that represents the given array a[] representation above (4.5 points).
2) What is **ONE** integer key that violates the max-heap property? (0.5 points) Explain why it violates the max-heap property (worth 5 points).

QUESTION 3 (20 POINTS)

Suppose that we are asked to implement hasCode() that computes the hash code in the Mystery class. We are considering two implementations—hashCode1() and hashCode2() as follows:

```
import java.util.Random; //used to generate random numbers in Java.

public class Mystery {
        public static final String UUID = "64de13c7fc79154d0570a12f268df";
        public Random random = new Random(); //initialize random variable

        //Other instance variables, constructors, and methods are not shown.

        private int hashCode1() {
                //compute and return the hash code.
        return UUID.hashCode(); //In this line of code, we use Java's built-in
hashCode() that we studied in class.
        }

        private int hashCode2() {
                //compute and return the hash code.
        return random.nextInt();//returns one random int value, which is produced
with approximately equal probability in the range of all possible $2^{32}$ possible int
values
        }
```

QUESTION 3-A) hashCode1() (worth 10 points)

QUESTION 3-B) hashCode2() (worth 10 points)

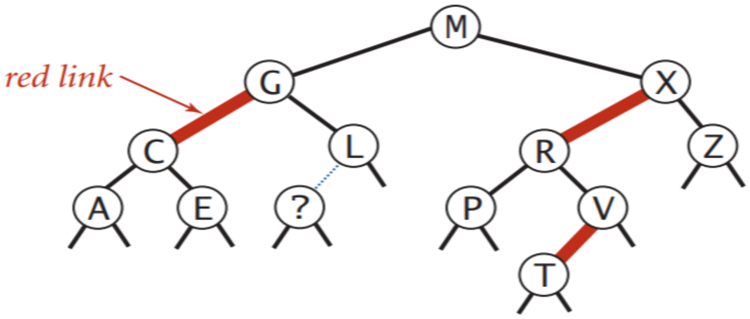
For **EACH** of those two hashCode() implementations—hashCode1(), hashCode2() above,

- Explain which of the primary requirements of hash functions that we studied in class is violated. If the implementation satisfies ALL the primary requirements of hash functions that we studied in class, explain how it satisfies those requirements.

⌘ Note: both implementations—hashCode1(), hashCode2() are considered to be efficient to compute.

QUESTION 4 (25 POINTS)

Consider left-leaning red-black (LLRB) trees that we studied in class. The following tree T is a left-leaning red-black tree that implements sorted map ADT. Red links are colored in red as shown below.



1) Which one or more of the keys below could be in the node labeled with a question mark (the left child of the node that stores L)? (8 points) Explain why (7 points).

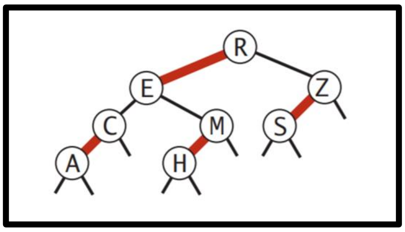
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2) What is the color of the link from the node labeled with a question mark to its parent? (i.e., link between ? and L) (worth 5 points) Explain why (worth 5 points).

QUESTION 5 (29.5 POINTS)

Give a sequence of 4 elementary operations (from phase 1, 2, 3, 4 in that order) that we perform when we insert the key **P** into the left-leaning red-black (LLRB) tree below.

1) At each phase, we perform one of the three elementary operations—color flip, left rotation, and right rotation. (worth 20 points)



	key	One of the three elementary operations
Phase 1		
Phase 2		
Phase 3		
Phase 4		

***Refer to the alphabetical order & examples (see the next page).**

2)

A. Draw a resulting left-leaning red-black (LLRB) tree after we insert the key **P** into the left-leaning red-black tree shown above in question 5-1) (4.5 points)

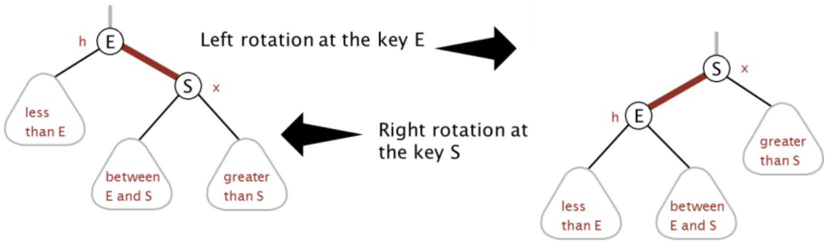
B. List the red links in the resulting LLRB tree after inserting the key **P** (5 points).

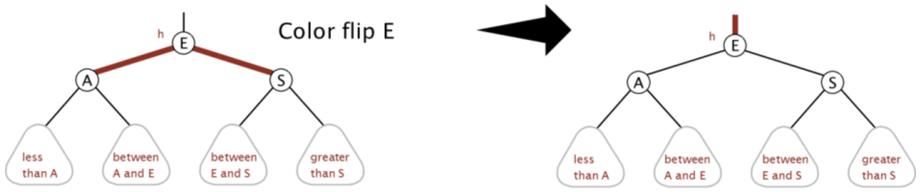
- To mark which one is the red link in the resulting LLRB tree, list ALL the red links clearly. (i.e., **You don't need to use a red pen.)**
    - For instance, for the tree shown in question 5-1) above, we list ALL the red links as follows**:**
        - Link between A and C
        - Link between E and R
        - Link between H and M
        - Link between S and Z

⌘ Refer to the alphabetical order below:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

⌘ For your reference, examples of color flips and tree rotations at the particular key are shown below:


Left rotation at the key E
Right rotation at the key S
h
E
S
x
less than E
between E and S
greater than S
S
x
h
E
greater than S
less than E
between E and S


Color flip E
h
E
A
S
less than A
between A and E
between E and S
greater than S
h
E
A
S
less than A
between A and E
between E and S
greater than S