CSE 12 SP24 Exam 3

Do not begin until instructed.

This exam is closed book, closed notes. No study aids or electronic devices are allowed. Turn off your phone and have no bags open.

You must hand in all sheets, including scratch paper, at the end of the exam.

This exam is designed to assess **your** understanding of topics from CSE12. You cannot communicate with other students during the exam. You cannot discuss this exam with anyone from now until you receive your grade on the exam. This includes posting about the exam on Piazza or elsewhere online.

Please write “I excel with integrity” in the box below:

Sign your name: _____

Print your name: _____

Print your email: _____

PID: _____

You have 45 minutes to complete the exam. Work to maximize points. **Write all your answers on the provided answer sheet on the back of this page.** If you get stuck, work through other problems, and come back to it.

Once the exam starts, you can rip off/remove the answer sheet.

In general, if you think you've spotted a typo in the exam, do your best to answer in the spirit of the question. Keep in mind that some questions have interesting code examples with intentional bugs for you to find as part of the question. If you ask a question during the exam, we may decline to answer. In general, assume that any necessary libraries (JUnit, ArrayList, List, and so on) have been imported.

Stay calm – you can do this!

Do not begin until instructed.

Write all answers below. There are 24 total points.

Question 1

Part 1 *Array contents after poll() (2 pts)*

Part 2-4 *(max heap 2 pts, runtime 2 pts, min heap 1 pt)*

1.2 Max Heap *(Write letter)*	**1.3 Runtime**	**1.4 Min Heap** *(Write letter)*

Question 2

Write letters of valid BSTs (2pts) / valid Max Heaps (2pts)

2.1 Valid BSTs	**2.2 Valid Max Heaps**

Question 3

Write sequence of letters (3 pts)

Next()

Question 4

Parts 1-5 *Write the letter (1 pt each)*

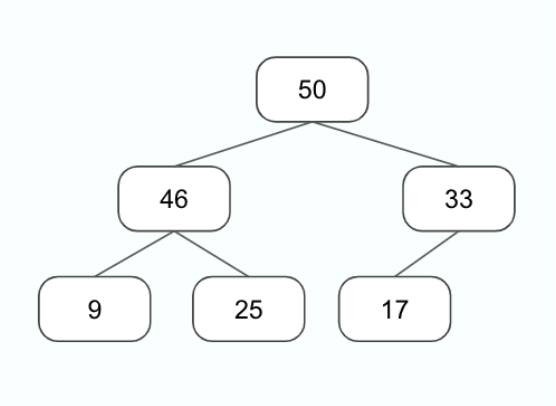
4.1	**4.2**	**4.3**	**4.4**	**4.5**

Part 6-7 *(BST 3 pts, Min Heap 2 pts)*

4.6 findMinimum() BST *(Write sequence of letters)*	**4.7 findMinimum() Min Heap** *(Write line of code)*

Question 1

1. Consider the following array representing a complete tree in **max-heap order**, along with a diagram illustrating its binary tree structure: `50, 46, 33, 9, 25, 17`



Assuming we use the typical strategy of moving the last element to the root and then using **bubbleDown()**, what is the resulting tree after a call to **poll()**? Write the **array's** contents directly in the answer sheet.

2. Assuming we use the typical strategy of adding an element at the end of the array and using **bubbleUp()**. Starting with an empty **max heap**, insert the following **five** elements in the given order: *1, 9, 3, 15, 6*. Write just **the letter** of the array that correctly represents the final **max heap**.

A. `15 3 9 1 6`
B. `15 3 6 9 1`
C. `15 3 9 6 1`
D. `15 9 6 3 1`
E. `15 9 3 1 6`
F. `15 9 3 6 1`

3. Consider the sequence of operations in the previous part (Question 1.2). What is the runtime of consecutively inserting elements (for a total of **n** number of elements) into an empty **max heap**? Give a Big-Theta bound, and write the answer in terms of **n**.

4. Consider the following array representing a complete tree in **min-heap order**: `5, 10, 8, 14`

Insert the element ***6*** into the array. Write just **the letter** of the array that correctly represents the final **min heap**.

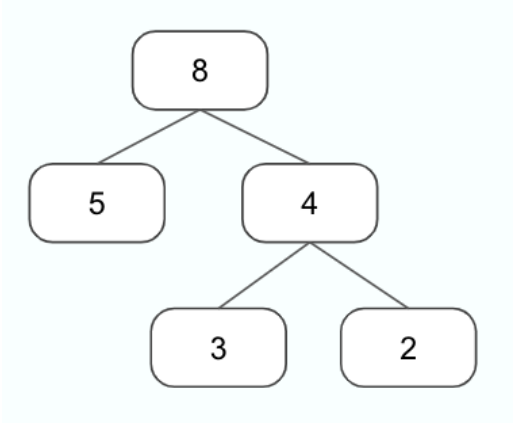
A. `5 8 10 6 14`
B. `5 8 6 10 14`
C. `5 8 6 14 10`
D. `5 6 8 14 10`
E. `5 6 8 10 14`
F. `5 6 10 8 14`

Question 2

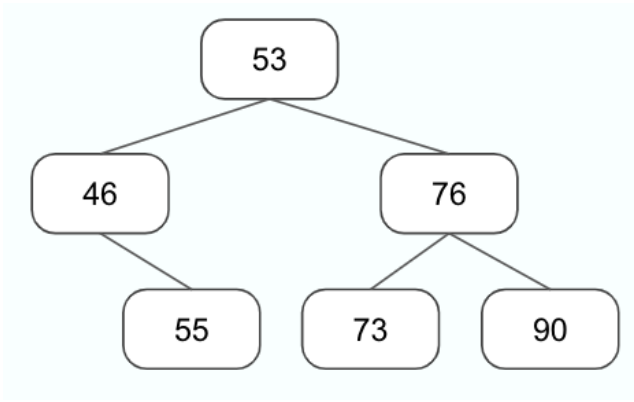
1. For each of the following trees, determine which ones are **valid** Binary Search Trees. Write the **letters** corresponding to the valid BSTs directly on the answer sheet

2. For each of the following trees, determine which ones are **valid** Max Heaps. Write the **letters** corresponding to the valid Max Heaps directly on the answer sheet

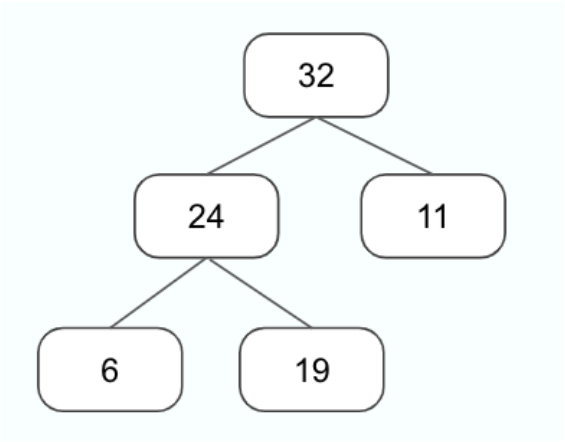
A.



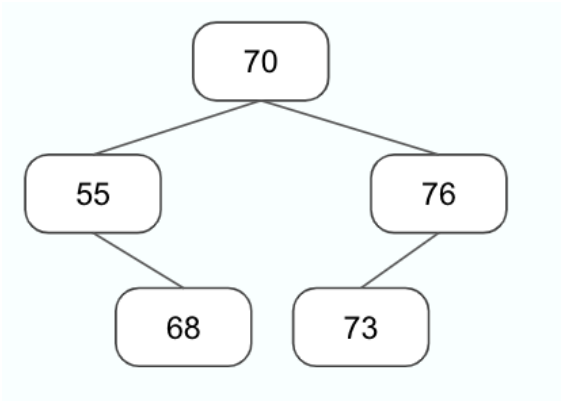
B.



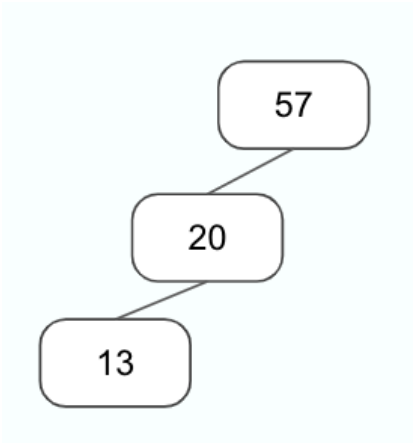
C.



D.



E.



F.

Question 3

Consider the following `IList` class extended to support iteration in the **reverse order**.

```
class IList implements Iterable<Integer> {
    class IListIterator implements Iterator<Integer> {
         int currentIndex;
         int size;
         IList ilist;
         public IListIterator(int startingIndex, int size, IList ilist) {
             this.currentIndex = startingIndex;
             this.size = size;
             this.ilist = ilist;
         }
         public Integer next() {
             /* Implement this */
         }
         public boolean hasNext() {
             return this.currentIndex >= 0;
         }
    }

    Integer[] elements;
    int size;
    public IList() {
        this.elements = new Integer[10];
        this.size = 0;
    }
    public Iterator<Integer> iterator() {
        return new IListIterator(this.size-1, this.size, this);
    }
    /* Other methods not shown (add, remove, etc.) */
}
```

Choose a sequence of letters from below that constructs a working implementation for the `next()` method that returns the elements of the list in order. Give your answer as a sequence of letters. You can ignore error cases of iterating past the end of the list.

A. `return ilist.elements[currentIndex];`
B. `currentIndex++;`
C. `currentIndex--;`
D. `return currentIndex;`
E. `Integer answer = ilist.elements[currentIndex];`
F. `return answer;`
G. `ilist.elements[currentIndex]--;`
H. `ilist.elements[currentIndex]++;`
I. `Integer answer = ilist.elements[currentIndex++];`
J. `Integer answer = ilist.elements[currentIndex--];`

Question 4

Assume two trees, a **Min Heap** and a **Binary Search Tree**.
For the questions below, answer with letters **A** for **heap**, **B** for **BST**, or **C** for **neither**.

1. Do the properties of one tree make it easier to determine whether the tree contains a specific value than doing the same in the other tree? If so, answer A or B. If not, answer C.

2. Would one tree structure ensure the minimum element can be accessed in constant time? If so, answer A or B. If not, answer C.

3. Would all the operations of one tree structure be directly impacted by the height of the tree? If so, answer A or B. If not, answer C.

4. Would one tree have a more efficient **average** case runtime for inserting a new Node into the tree than the other tree? If so, answer A or B. If not, answer C.

5. Would one tree structure be more efficiently implemented in an array-based representation than the other tree structure? If so, answer A or B. If not, answer C.

A **Binary Search Tree** has the ability to efficiently find the minimum element of the tree.

6. Assume the Node<K,V> class and the Binary Search Tree class from PA7. Choose from the answer choices below to implement a method that finds the **value** with the **minimum key** in a Binary Search Tree, with correct syntax and semicolons/braces. Assume root is not null.

```
public V findMinimum() {
  // Implement this!

}
```

   A. `Node current = root;`
   B. `Node<K,V> current = root;`
   C. `Node<K,V> current = root.right;`
   D. `while (current != null) {`
   E. `while (current.right != null) {`
   F. `while (current.left != null) {`
   G. `current = current.left;`
   H. `current = current.right;`
   I. `return current.key;`
   J. `return current.value;`
   K. `return current;`
   L. `}`

7. Now assume we are writing the same method, `findMinimum()`, for a **Min Heap**. Write a **one-line** return statement that would provide such functionality. Note that `findMinimum()` should **NOT** modify the tree.

Scratch Paper