44-542 Object Oriented Programming

Exam 3 Version A (100 pts)

Problem 1 refers to the class Dog defined below.

public class Dog implements Comparable<Dog>

{

private String name;

private int age;

public Dog(String name, int age)

{

this.name = name;

this.age = age;

}

public String getName()

{

return name;

}

public int getAge()

{

return age;

}

@Override

public int compareTo(Dog otherDog)

{

return 0;

}

}

1. (10 pts) Rewrite the code for the compareTo method in the Dog class, assuming you want to compare Dog objects on the basis of their names. This will be the natural order for the Dog class and will result in Dog objects being ordered alphabetically by name in ascending order.

public int compareTo(Dog otherDog)

{

return this.name.compareTo(otherDog.name);

}

2. (6 pts) Assume we construct the binary search tree that results when the values are inserted in the order given:

12, 18, 3, 5, 22, 10, 13, 17, 30, 1

(a) What value is stored in the root node? 12

(b) What is the height of the tree? 3

(c) List all descendants of 3. 1, 5, 10

3. (6 pts) Assume we remove 17 from the binary search tree shown below.



AFTER the node is removed,

(a) What is the height of the tree? 4

(b) What value is stored at the root node? 18

(c) What value is stored at the right child of the root? 25

4. (6 pts) Draw the list produced by the code below.

myList = new LinkedList<Integer>();

myList.add(3);

myList.add(4);

myList.addLast(5);

myList.addFirst(6);

myList.addFirst(7);

myList.addLast(8);

7 → 6 → 3 → 4 → 5 → 8

5. (5 pts) Find the output of the following code.

LinkedList<String> dogs = new LinkedList<String>();

dogs.add("A");

dogs.add("B");

dogs.add("C");

dogs.add("D");

dogs.add("E");

ListIterator<String> itr = dogs.listIterator();

System.out.print(itr.next() + " ");

System.out.print(itr.next() + " ");

System.out.print(itr.previous() + " ");

System.out.print(itr.next() + " ");

System.out.println(itr.previous());

OUTPUT

A B B B B

6. (7 pts) Suppose we are using the hash function **h(key) = key MOD 7** to store records with keys **10, 11, 17, 2, 30**. Show where each key is stored in the table. Use linear probing to resolve collisions. You must insert the keys in the order in which they are listed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 2 | 10 | 11 | 17 | 30 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

7. (10 pts) Find the output of the code below.

public static int f(int n)

{

if (n == 0)

{

return 0;

}

else if (n % 2 == 0)

{

return f(n / 2);

}

else

{

return 1 + f(n / 2);

}

}

public static void main(String[] args) {

System.out.println(f(2));

System.out.println(f(3));

System.out.println(f(7));

System.out.println(f(12));

System.out.println(f(32));

}

OUTPUT

1

2

3

2

1

8. (6 pts) Find the output of the following code:

Stack<String> myStack = new Stack<String>();

myStack.push("I");

# OUTPUT

Python

and

Java

like

myStack.push("like");

myStack.push("Java");

myStack.push("and");

myStack.push("Python");

System.out.println(myStack.peek());

myStack.pop();

System.out.println(myStack.peek());

myStack.pop();

System.out.println(myStack.peek());

myStack.pop();

System.out.println(myStack.peek());

9. (4 pts) Which of the following is/are true of queues? Circle all true statements.

(a) all elements are inserted at the front of a queue

(b) all elements are removed at the front of a queue

(c) queues are FIFO structures

(d) queues are LIFO structures

10. (6 pts) Find the output of the code below:

TreeSet<Integer> numSet = new TreeSet<Integer>();

numSet.add(10);

numSet.add(7);

OUTPUT

4

4

7

10

20

numSet.add(20);

numSet.add(4);

numSet.add(7);

numSet.add(20);

System.out.println(numSet.size());

for(int x: numSet)

{

System.out.println(x);

}

11. (10 pts) Find the output of the code below:

TreeMap<String,Integer> ages

= new TreeMap<String,Integer>();

ages.put("Midge", 11);

ages.put("Carmen", 8);

ages.put("Lucy", 4);

ages.put("Leonard", 3);

System.out.println(ages.get("Leonard"));

System.out.println(ages.keySet());

System.out.println(ages.values());

System.out.println(ages.entrySet());

OUTPUT

3

[Carmen, Leonard, Lucy, Midge]

[8, 3, 4, 11]

[Carmen=8, Leonard=3, Lucy=4, Midge=11]

12. (3 pts) What is the minimum height of a binary tree with 13 elements?

(a) 1

(b) 2

(c) 3

(d) 4

(e) 5

13. (12 pts) For the tree below



(a) List the entries in the tree in the order they are visited in a *preorder* traversal.

A B D E C F

(b) List the entries in the tree in the order they are visited in a *breadth first* traversal.

A B C D E F

14. (3 pts) For binary search trees, the search time is always logarithmic in n.

(a) True

(b) False

15. (6 pts) Assume we rotate right around 32 in the binary search tree shown below.



AFTER the rotation,

(a) What value is stored at the root of the tree? 25

(b) What is the height of the tree? 3

(c) What value is stored in the right child of the root? 32