Final exam
Study guide

Algorithms:

- Convert the following expressions to postfix

  (3 + 4) / (12 + 13 / 7)

  A + b * c * (d - e + f)

- Evaluate the following postfix expression

  12 4 + 6 6 2 / + *

- Draw the in-fix expression tree for the following expression. Write out its post-fix version using the tree.

  a – b / ((c * d) + e)

- Create a binary tree from the following array of values. Draw the tree.

11	17	21	42	66	78	96

- What is the time complexity of inserting an element into a BST?

- If a set of elements is inserted into a BST in two different orders:
  - will the two corresponding BSTs look the same?
  - Will the inorder traversal be the same?
  - Will the postorder traversal be the same?
  - Will the preorder traversal be the same?

- Perform the following operations sequentially on a BSTree that is initially empty.
  - insert 10, 4, 2, 20, 8, 9, 15, 13, 50
  - delete 20, 2, 50, 15, 10, 4

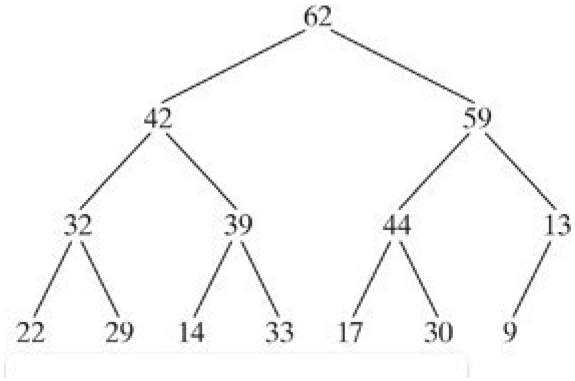
- create a min/max-heap binary tree of the following array: (as binary tree)

12	90	60	45	25	100	36	85

- insert 88 in the above max-heap
- perform remove operation on above max-heap

- Given the following heap, show the steps of removing all nodes from the heap.

Final exam
Study guide



- What is the height of a nonempty heap? What is the height of a heap with 16, 17, and 512 elements? If the height of a heap is 5, what is the maximum number of nodes in the heap?

Which of the data types below does not allow duplicates?
a. Set
b. List
c. Vector
d. Stack
e. LinkedList

Which of the data types below could be used to store elements in their natural order based on the compareTo method?
a. HashSet
b. TreeSet
c. LinkedHashSet

d. Collection

e. Set

What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<A>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A  {
  int r = 1;

  public String toString() {
    return r + "";
  }

  public boolean equals(Object o) {
    return this.r == ((A)o).r;
  }

  public int hashCode() {
    return r;
  }
}
```

a. [1]

b. [1, 1]

c. [1, 1, 1]

d. [1, 1, 1, 1]

Final exam
Study guide

What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A  {
  int r = 1;

  public String toString() {
    return r + "";
  }

  public int hashCode() {
    return r;
  }
}
```

a. [1]
b. [1, 1]
c. [1, 1, 1]
d. [1, 1, 1, 1]

What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A
  { int r =
  1;

  public String toString() {
    return r + "";
  }

  public boolean equals(Object o) {
    return this.r == ((A)o).r;
  }
}
```

a. [1]
b. [1, 1]
c. [1, 1, 1]
d. [1, 1, 1, 1]

Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.addAll(s2), s1 is __________.

a. [1, 2, 2, 3, 5, 6]
b. [1, 2, 3, 5, 6]
c. [1, 5]
d. [2]

Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.removeAll(s2), s1 is __________.

a. [1, 2, 2, 3, 5, 6]
b. [1, 2, 3, 5, 6]
c. [1, 5]
d. [2]

- Binary tree time complexity
- In-order, pre-order, post-order BST traversal, algorithm (recursive, non-recursive)
- BreadthFirst traversal
- Height of BST
- Number of nodes, leaf-nodes of a BST
- Complete binary tree representation
- Heap and PQ
- Hashing
- Set and Map operations