

**Computer Science 3 Final Review**

1. **What is the best case for finding an item in a binary search tree?**

2. **What type of tree traversal is this?**

```
private void traverse(TreeNode tree){  
    if (tree != null){  
        traverse(tree.getRight());  
        out.print(tree.getValue() + " ");  
        traverse(tree.getLeft());  
    }  
}
```

3. **A complete binary tree will**

- A) have all levels full
- B) have all levels full that can be and all partial levels shifted to the left
- C) have all levels shifted to the left
- D) never have leaves
- E) be balanced

4. **Which of the following is true for a full binary tree?**

- A) the bottom most level has mostly leaves
- B) the bottom most level is almost complete
- C) the bottom most level is nothing but parents
- D) the bottom most level has the biggest nodes
- E) the bottom most level has nothing but leaves

5. **How many nodes would a full tree with 5 levels have?**

**6. What is the output of the following code?**

```
TreeNode w = new TreeNode(90,  
    new TreeNode(100,  
        new TreeNode(13,null,null),  
        new TreeNode(45,null,null)),  
    new TreeNode(200,  
        new TreeNode(99,null,null),  
        new TreeNode(13,null,null)));  
  
out.println(w.getLeft().getLeft().getValue());
```

**7. If you insert the following numbers in the order listed into a binary search tree, which of the following represents the tree's post order traversal output?**

**Items inserted in the following order : 200 150 225 75 210 250**

**8. How many leaves max could a complete tree with 6 nodes have?****9. Which tree traversal algorithm prints all the values of a binary search tree out in ascending order?****10. Which Java class is an example of a heap?**

- A) ArrayList
- B) LinkedList
- C) TreeMap
- D) TreeSet
- E) PriorityQueue

**11. What is the run time of inserting into a heap?**

Name: \_\_\_\_\_

ID: A

12. **What is the run time of building a heap?**

13. **What is output by the code below?**

```
LinkedList<String> bob;  
bob = new LinkedList<String>();  
bob.add( "aplus" );  
bob.addFirst( "1" );  
bob.addFirst( "comp" );  
bob.add( 0, "7" );  
System.out.println(bob);
```

14. After running the code below, how many values would the linked list at spot 3 contain?

```
public class HashTable
{
    private int size;
    private ListNode[] table;

    public HashTable(int numSlots) {
        size = numSlots;
        table = new ListNode[size];
    }

    public void add(Object obj)
    {
        ListNode element = new ListNode(obj, null);
        int index = obj.hashCode() % size;

        if(table[index] == null)
        {
            table[index] = element;
        }
        else
        {
            ListNode current = table[index];
            while(current.getNext() != null)
                current = current.getNext();
            current.setNext(element);
        }
    }
}

//client code
HashTable h = new HashTable(10);
int[] nums = {10,7,8,13,11,4,5,1,12,5,6,9,2,3,17,27,31,35};
for( int val : nums )
{
    h.add( val );
}
```

**15. What is output by the code below?**

```
LinkedList<String> ann;  
ann = new LinkedList<String>();  
ann.add("3");  
ann.add(0, "0");  
ann.add("2");  
ListIterator x = ann.listIterator();  
x.next();  
x.add( "dog" );  
x.add( "cat" );  
System.out.println(ann);
```

**16. When referring to hash functions what is a collision?**

- A) When a single item has multiple different hashes
- B) The slot in the hash table an item hashes to
- C) When a hash table runs out of space
- D) When two keys hash to the same slot
- E) When duplicate items hash to different slots

**17. How does chaining or using buckets resolve collisions?****18. What is the difference between a singly-linked list and a doubly-linked list?**

19. Which of the following best matches the running time of accessing an element in these two lists?

	array	LinkedList
LINE 1	O(1)	O(1)
LINE 2	O(1)	O(N)
LINE 3	O(1)	O(N <sup>2</sup> )
LINE 4	O(N)	O(N)
LINE 5	O(N)	O(N <sup>2</sup> )

- A) Line 1
- B) Line 2
- C) Line 3
- D) Line 4
- E) Line 5

20. Which of the following is a child of SortedMap?

- A) ArrayList
- B) HashMap
- C) TreeMap
- D) Map
- E) LinkedList

21. What is output by the code below?

```
Map<String,Integer> map;  
map = new TreeMap<String,Integer>();  
map.put("6",81);  
map.put("3",3);  
map.put("7",213);  
map.put("5",9);  
map.put("1",876);  
  
out.println(map.put("5",1));
```

22. What is the output by the code below?

```
Map<Integer,Integer> map;  
map = new TreeMap<Integer,Integer>();  
Integer[] list = {4,5,8,2,2,2,3,3,6,6};  
  
for(Integer num : list)  
    if(map.get(num)==null)  
        map.put(num,1);  
    else  
        map.put(num,map.get(num)+1);  
  
out.println(map);
```

23. Which of the following interfaces must be implemented by all Objects stored in a TreeMap?

- A) Locatable
- B) Sortable
- C) Treeable
- D) Mapable
- E) Comparable

24. What is output by the code below?

```
TreeMap<Integer,Integer> map;  
map = new TreeMap<Integer,Integer>();  
Integer[] list = {9,2,3,4,5,6,7,10,3,4,2,3,4,2,3};  
  
for(Integer num : list)  
    if(map.get(num)==null)  
        map.put(num,1);  
    else  
        map.put(num,map.get(num)+1);  
  
System.out.println( map.keySet() );
```

**25. What is output by the code below?**

```
Stack<Integer> a = new Stack<Integer>();
Queue<Integer> b;
b = new LinkedList<Integer>();
a.push(5);
a.push(7);
b.add(a.pop());
b.add(a.pop());
out.println(b);
```

**26. What is the output by the code below?**

```
String s = "rstuv";
Queue<Character> q;
q = new LinkedList<Character>();
for(char let : s.toCharArray())
{
    q.add(let);
}
q.remove();
q.add('w');
q.add('x');
q.add('y');
q.add('z');
q.remove();
out.println(q.remove());
```



**27. What is output by the code below?**

```
String word = "guxys";
PriorityQueue<Character> theQ;
theQ = new PriorityQueue<Character>();
for(char ch : word.toCharArray())
{
    theQ.add(ch);
}
theQ.remove();
theQ.remove();
theQ.remove();
theQ.remove();
theQ.add('r');
theQ.add('t');
theQ.add('z');
theQ.add('y');
theQ.remove();
out.println(theQ.remove());
```

**28. What is output by the code below?**

```
PriorityQueue<String> pQ;
pQ = new PriorityQueue<String>();
pQ.add("d");
pQ.add("a");
pQ.remove();
pQ.add("z");
pQ.add("w");
pQ.add("b");
pQ.remove();
out.println(pQ);
```

29. Which of the following statements would correctly fill <\*1>?

```
class AplusQ {
    Object[] stuff;
    int head, tail;

    public AplusQ() {
        this(10);
    }

    public AplusQ(int l) {
        stuff = new Object[l];
        head = 0;
        tail = 0;
    }

    public boolean enqueue(Object x) {
        if ( <*1> ) // overflow
            return false;
        stuff[tail++ % stuff.length] = x;
        return true;
    }

    public Object dequeue() {
        if ( <*2> ) // empty queue
            return null;
        return stuff[head++ % stuff.length];
    }
}
```

- A) `tail - stuff.length <= head`
- B) `tail - stuff.length < head`
- C) `tail - stuff.length >= head`
- D) `tail - stuff.length > head`
- E) `tail >= head`

30. Which of the following is an interface?

- A) `ArrayList`
- B) `HashSet`
- C) `TreeMap`
- D) `Collection`
- E) `LinkedList`

31. What is the output of the code below?

```
Set<Integer> s = new TreeSet<Integer>();  
s.add( 31 );  
s.add( 65 );  
s.add( -87 );  
System.out.println( s );
```

32. What is the output by the code below?

```
Set<String> s = new TreeSet<String>();  
s.add("one");  
s.add("two");  
s.add("three");  
out.println(s);
```

33. What is the output by the code below?

```
Set<String> s = new TreeSet<String>();  
s.add("one");  
s.add("two");  
s.add("one");  
s.add("two");  
out.println(s);
```

34. Which of the following best matches the runtime for HashSet?

	add()	contains()
Line 1.	$O(1)$	$O(N)$
Line 2.	$O(1)$	$O(1)$
Line 3.	$O(N)$	$O(\log_2 N)$
Line 4.	$O(\log_2 N)$	$O(\log_2 N)$
Line 5.	$O(\log_2 N)$	$O(N)$

- A) Line 1
- B) Line 2
- C) Line 3
- D) Line 4
- E) Line 5

35. Which of the following best matches the runtime for TreeSet?

	add()	contains()
Line 1.	$O(1)$	$O(N)$
Line 2.	$O(1)$	$O(1)$
Line 3.	$O(N)$	$O(\log_2 N)$
Line 4.	$O(\log_2 N)$	$O(\log_2 N)$
Line 5.	$O(\log_2 N)$	$O(N)$

- A) Line 1
- B) Line 2
- C) Line 3
- D) Line 4
- E) Line 5

36. What is the resulting set?

$$A = \{1, 4, 5, 7\}, B = \{2, 4, 5, 9\}$$

$$A \cup B$$

Name: \_\_\_\_\_

ID: A

37. What is the resulting set?

$$A = \{1, 4, 5, 7\}, B = \{2, 4, 5, 9\}$$

$$A \cap B$$

38. What is the resulting set?

$$A = \{1, 4, 5, 7\}, B = \{2, 4, 5, 9\}, C = \{1, 2, 7, 9\}$$

$$(A - B) \cap C$$

39. What is the resulting set?

$$A = \{1, 4, 5, 7\}, B = \{2, 4, 5, 9\}, C = \{1, 2, 7, 9\}$$

$$A \cup (B \cap C)$$

40. What does it mean when two sets are disjoint?

41. What is the cardinality of a set?

42. What is the output by the code below?

```
Integer[] ar = {51, 42, 21, 13},  
          br = {17, 42, 13, 9, 100};  
Set<Integer> as = new TreeSet<Integer>(Arrays.asList(ar)),  
            bs = new TreeSet<Integer>(Arrays.asList(br)),  
            cs = new TreeSet<Integer>(as);  
cs.retainAll(bs);  
out.println(cs);
```

43. What is the output by the code below?

```
Integer[] ar = {51, 42, 21, 13},  
          br = {17, 42, 13, 9, 100};  
Set<Integer> as = new TreeSet<Integer>(Arrays.asList(ar)),  
            bs = new TreeSet<Integer>(Arrays.asList(br)),  
            cs = new TreeSet<Integer>(as);  
cs.removeAll(bs);  
out.println(cs);
```

44. Set A is a subset of set B if?

- A) The max value in set A is less than the max value in set B
- B) The sum of the elements in set A is less than the sum of the elements in set B
- C) All elements in set A are in set B
- D) Set A has a lesser cardinality than set B
- E) None of the elements in set A are in Set B

45. What sort has a partition method that uses a pivot location?

46. What is the bigO of the code below?

```
int n = //user input
for(int i=0; i<n; i++){
    for(int j=1; j<n; j=j*2){
        System.out.println(i*j);
    }
}
```

47. What is the BigO for searching a single linked linked-list?

48. Which of these is the correct BigO for adding an item to the front of an array?

49. Which of these is the correct BigO for adding an item to the front of a Java LinkedList?

50. Assuming <blank 1> is filled correctly, what sort is sortOne()?

```
public static void sortOne( Comparable[] list )
{
    for(int i=0; i<list.length-1; i++)
    {
        int min = i;
        for(int j=i+1; j<list.length; j++)
        {
            if(list[j]. < blank 1 > (list[min])<0)
                min = j;
        }
        if( min != i)
        {
            Comparable temp = list[min];
            list[min] = list[i];
            list[i] = temp;
        }
    }
}
```

51. Which sort splits data into smaller lists, sorts the smaller lists, and then combines all of the sorted smaller lists back into one big list?



52. What would be the value of list after four passes of the `help()` method?

```
public void sort( int[] list, int front, int back)
{
    int mid = (front+back)/2;
    if( mid==front) return;
    sort(list, front, mid);
    sort(list, mid, back);
    help(list, front, back);
}
```

```
private void help(int[] list, int front, int back)
{
    int[] temp = new int[back-front];
    int i = front, j = (front+back)/2, k =0;
    int mid =j;
    while( i<mid && j<back)
    {
        if(list[i]< list [j])
            temp[k++] = list[i++];
        else
            temp[k++] = list[j++];
    }

    while(i<mid)
        temp[k++] = list[i++];
    while(j<back)
        temp[k++] = list[j++];
    for(i = 0; i<back-front; ++i)
        list[front+i]=temp[i];
}
```

**//code in the main**

```
int[] list = {39,6,11,23,18,3,20,5,57};
sort(list, 0, list.length);
```

53. How many times would `help()` be called before list was sorted?

```
public void sort( int[] list, int front, int back)
{
    int mid = (front+back)/2;
    if( mid==front) return;
    sort(list, front, mid);
    sort(list, mid, back);
    help(list, front, back);
}

private void help(int[] list, int front, int back)
{
    int[] temp = new int[back-front];
    int i = front, j = (front+back)/2, k =0;
    int mid =j;
    while( i<mid && j<back)
    {
        if(list[i]< list [j])
            temp[k++] = list[i++];
        else
            temp[k++] = list[j++];
    }

    while(i<mid)
        temp[k++] = list[i++];
    while(j<back)
        temp[k++] = list[j++];
    for(i = 0; i<back-front; ++i)
        list[front+i]=temp[i];
}

//code in the main
int[] list = {39,6,11,23,18,3,20,5,57};
sort(list, 0, list.length);
```

**54. Which sort is shown below?**

```
public void sort( int[] list, int front, int back)
{
    int mid = (front+back)/2;
    if( mid==front) return;
    sort(list, front, mid);
    sort(list, mid, back);
    help(list, front, back);
}

private void help(int[] list, int front, int back)
{
    int[] temp = new int[back-front];
    int i = front, j = (front+back)/2, k =0;
    int mid =j;
    while( i<mid && j<back)
    {
        if(list[i]< list [j])
            temp[k++] = list[i++];
        else
            temp[k++] = list[j++];
    }

    while(i<mid)
        temp[k++] = list[i++];
    while(j<back)
        temp[k++] = list[j++];
    for(i = 0; i<back-front; ++i)
        list[front+i]=temp[i];
}
```

**55. What is output by the code below?**

```
ArrayList<Integer> x = new ArrayList<Integer>();
x.add(11);
x.add(18);
x.add(9);
x.add(22);
Iterator<Integer> it = x.iterator();
it.next();
System.out.println(it.next());
```

**56. What is output by the code below?**

```
ArrayList<Integer> z;  
z = new ArrayList<Integer>();  
z.add(11);  
z.add(18);  
z.add(89);  
z.add(22);  
z.add(3);  
Iterator<Integer> itera = z.iterator();  
while(itera.hasNext()){  
    if(itera.next().compareTo(89)==0)  
        itera.remove();  
}  
System.out.println(z);
```

**57. What is output by the code below?**

```
ArrayList<Integer> a = new ArrayList<Integer>();  
a.add(11);  
a.add(18);  
a.add(80);  
a.add(22);  
a.add(3);  
ListIterator<Integer> iterator = a.listIterator();  
iterator.next();  
iterator.next();  
iterator.set(99);  
iterator.next();  
iterator.next();  
iterator.previous();  
iterator.set(33);  
System.out.println(a);
```

58. What is output by the code below?

```
Stack<Integer> a = new Stack<Integer>();
Queue<Integer> b;
b = new LinkedList<Integer>();
a.push(5);
a.push(7);
b.add(a.pop());
b.add(a.pop());
out.println(b);
```

59. Attempting to pop from an empty stack results in?

60. Assuming all the methods in this stack are properly implemented what is the running time of pop?

```
public class Stack<E> {
    public boolean empty();
    public E peek();
    public E pop();
    public E push(E item);
}
```

61. What is output by the code below?

```
public String fun (String str) {
    Stack<Character> a = new Stack<Character>();
    String res = "";
    for (char b: str.toCharArray()) {
        a.push(b);
    }
    while (!a.empty()) {
        res += a.pop();
    }
    return res;
}
```

**// client code**

```
out.println(fun("Hello, World!"));
```

**62. What is output by the code below?**

```
Stack<Integer> s = new Stack<Integer>();
s.push(70);
s.push(25);
s.push(42);
out.println(s.search(70));
```

**63. What is output by the code below?**

```
public static Stack<Integer> fun (Stack<Integer> aplus)
{
    Stack<Integer> comp = new Stack<Integer>();
    Stack<Integer> ans = new Stack<Integer>();

    while (!aplus.empty())
    {
        int m = aplus.peek();
        while (!aplus.empty())
        {
            if (aplus.peek() < m)
                m = aplus.peek();
            comp.push(aplus.pop());
        }
        while (!comp.empty())
        {
            if (comp.peek() == m)
                ans.push(comp.pop());
            else
                aplus.push(comp.pop());
        }
    }
    return ans;
}

// client code
Stack<Integer> s = new Stack<Integer>();
int[] x = {-6,20,111,-87,300,5};
for( int item : x )
    s.add( item );
System.out.println(fun(s));
```

**64. What does method fun do?**

```
public static Stack<Integer> fun (Stack<Integer> aplus)
{
    Stack<Integer> comp = new Stack<Integer>();
    Stack<Integer> ans = new Stack<Integer>();

    while (!aplus.empty())
    {
        int m = aplus.peek();
        while (!aplus.empty())
        {
            if (aplus.peek() < m)
                m = aplus.peek();
            comp.push(aplus.pop());
        }
        while (!comp.empty())
        {
            if (comp.peek() == m)
                ans.push(comp.pop());
            else
                aplus.push(comp.pop());
        }
    }
    return ans;
}
```

## 65. What is output by the code below?

```
public static int fun (String str)
{
    String aplus = "{[()]}" ;
    int res = 0, center = aplus.length() / 2;
    Stack<Character> comp = new Stack<Character>();
    for (char x: str.toCharArray())
    {
        if (aplus.contains("" + x))
        {
            if (aplus.indexOf(x) < 3)
            {
                comp.push(x);
                if (comp.size() > res)
                    res = comp.size();
            }
            else
            {
                int ind = aplus.indexOf(x);
                int ref = center - (ind - center + 1);
                if (comp.empty() ||
                    comp.pop() != aplus.charAt(ref))
                    return -1;
            }
        }
    }
    return res;
}

// client code
out.println(fun("{() ()} ([[]{}]) ()"));
```