Points off	1	2	3	4	5		Total off	Net Score

Your Name: _____

Your UTEID: _____

Circle your TA's Name: **Anthony Ashley David K. David T. Grace Henry Lilly Neal Nina Noah Pranav Skyler Sam**

Instructions:

1. There are **5** questions on this exam. 100 points available. Scores shall be scaled to 250 points.
2. You have 3 hours to complete the exam.
3. Place your final answers on this exam. Not on the scratch paper. **Answer in pencil**.
4. You may not use a calculator or **outside resources of any kind** while taking this exam.
5. When answering coding questions, ensure you follow the restrictions of the question.
6. Do not write code to check the preconditions.
7. On coding questions, you may implement your own helper methods.
8. On coding questions make your solutions as efficient as possible given the restrictions of the question.
9. Exam proctors will not answer any questions regarding the content of the exam. If you think a question is ambiguous or has an error, state your assumptions and answer based on those assumptions.
10. When you complete the exam show the proctor your UTID, give them the exam and all the scratch paper, used or not, and leave the room quietly.

1. (2 points each, 50 points total) Short answer. Place your answer on the line next to or under the question. Assume all necessary imports have been made.

   a. If a question contains a syntax error or compile error, answer **compile error**.
   b. If a question would result in a runtime error or exception, answer **runtime error**.
   c. If a question results in an infinite loop, answer **infinite loop**.
   d. Recall when asked for Big O your answer shall be the most restrictive correct Big O function. Give the most restrictive, correct Big O function. (Closest without going under.)
   e. Assume $\log_2(1{,}000) = 10$ and $\log_2(1{,}000{,}000) = 20$.

A. Circle the sorting algorithm that has the best best-case order (Big O) when implemented as shown in class.

Selection Sort Insertion Sort Quick Sort Mergesort

B. Consider the following four trees. Assume each tree has the same number of nodes. Circle the one that is most likely to have the smallest height.

- a **Huffman Code** Tree for a typical, large English text
- a **maxheap**
- a **binary search tree** that uses the simple add algorithm and formed by inserting random data
- a **Red-Black Tree** formed by inserting data in sorted descending order

C. What is the maximum height of a Huffman Code Tree with 5 leaves?

_____

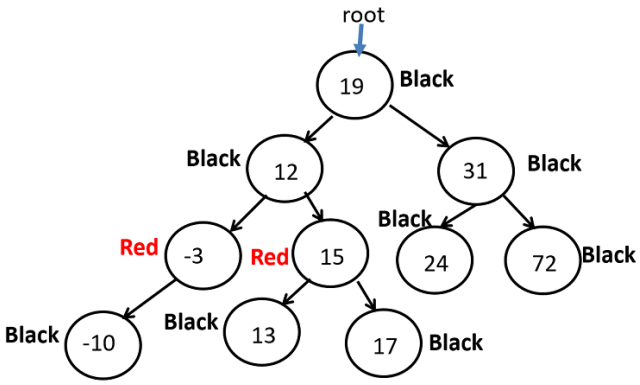
D. The following method takes 4 seconds to run when n = 1,000,000. **What is the expected time for the method to complete when n = 3,000,000?** Assume the Random constructor and nextInt() methods are O(1).

_____

```
public static List<Integer> getList2(int n) {
    Random r = new Random();
    ArrayList<Integer> result = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        int x = r.nextInt();
         if (x % 3 == 0)
             result.add(0, x); // position, val to add;
         else
             result.add(x);
    }
    return result;
}
```

E. Consider the tree to the right:

It does not meet the requirements for a Red-Black tree. Why not?



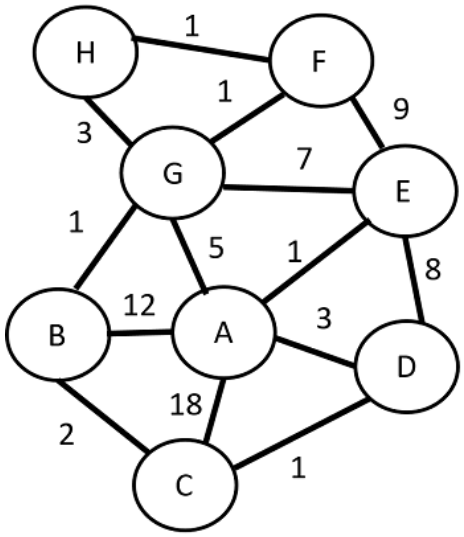
_____

F. What is output by the following code?

_____

```
Map<String, ArrayList<Integer>> tmap = new TreeMap<>();
ArrayList<Integer> list = new ArrayList<>();
list.add(3);
list.add(1);
tmap.put("A", list);
list.add(4);
tmap.put("N", list);
list = new ArrayList<>();
tmap.put("C", list);
list.add(7);
list.add(10);
tmap.put("J", list);
System.out.println(tmap);
```

G. Consider the undirected graph to the right:



We apply the shortest path algorithm demonstrated in class and implemented on assignment 11 to the graph with H as the starting vertex.

Which vertex will be the **fifth** vertex, including H itself, to have its scratch variable set to true indicating the algorithm has found the shortest path to that vertex?

_____

H. Suppose we are creating a new file format to represent information about college students. Each student record must store the college or university the student is currently attending. Assume students can attend exactly one college or university. They cannot attend multiple colleges or universities at the same time.

There are 3,000 possible college and universities the student could be attending.

If our goal is to minimize the number of bits in individual student record files used to represent the college or university the student is attending, what is the minimum number of bits necessary to encode the 3,000 possible college and universities?

_____

I. The following method takes 10 seconds to complete when n = 1,000,000. **What is the expected time for the method to complete when n = 4,000,000?** The binary search tree class use the simple add method demonstrated in class. Assume the Random constructor and nextInt() methods are O(1).

_____

```
public static BinarySearchTree<Integer> make2(int n) {
    BinarySearchTree<Integer> result;
    result = new BinarySearchTree<>();
    Random r = new Random();
    for (int i = 0; i < n; i++) {
        int val = r.nextInt() % 10;
        result.add(val);
    }
    return result;
}
```

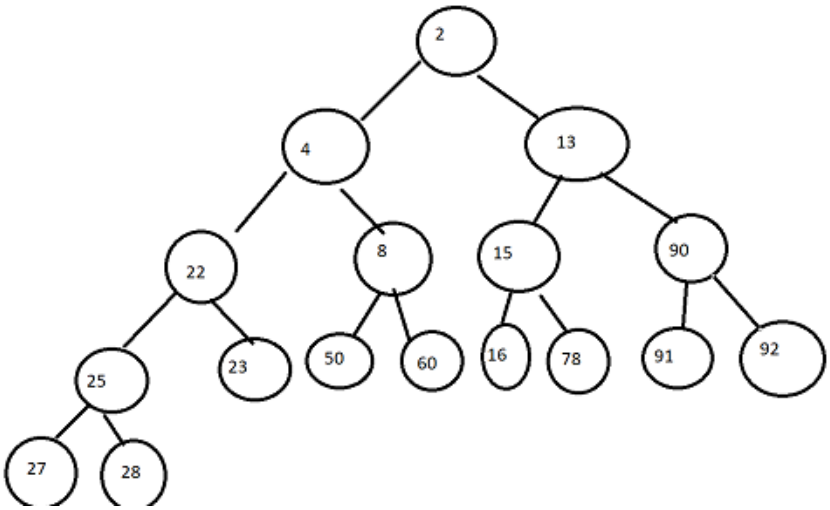
J. Are most real-world graphs sparse, dense, or is the number of dense graphs roughly equal to the number of dense graphs (close to a 50-50 split)? Circle the correct answer:

Sparse Dense Roughly Equal Split

K. Consider the heap to the right.

How many swaps will be performed when adding 4 to the heap using the algorithm demonstrated in class?

_____



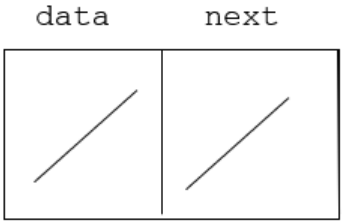
L. To the right is a graph represented by an adjacency matrix.

Is the graph directed or undirected and is it cyclic or acyclic?

_____

	A	B	C	D	E	F	G
A	0	1	1	0	1	0	0
B	1	0	0	0	1	0	0
C	1	0	0	0	1	0	1
D	0	0	0	0	0	0	0
E	1	0	1	1	0	0	0
F	0	1	0	0	0	0	0
G	0	0	1	1	0	0	0

M. Consider a singly linked node class like the one used in lecture when we developed our singly linked list. This Node class does not use Java generics. Each node has a reference to an object and another Node.



Consider the following code. The next reference in each Node is a public variable of type Node. The zero-argument constructor sets the data and next instance variables to null. The constructor that accepts an argument of type Node sets the new Node's next reference to the given Node and data to null.

Draw the variables, references, and objects that exist after the following code executes. Draw node objects as shown above and boxes for variables. The example has all instance variables set to `null`.

```
Node n1 = new Node();
n1.next = n1;
Node n2 = new Node(n1);
```

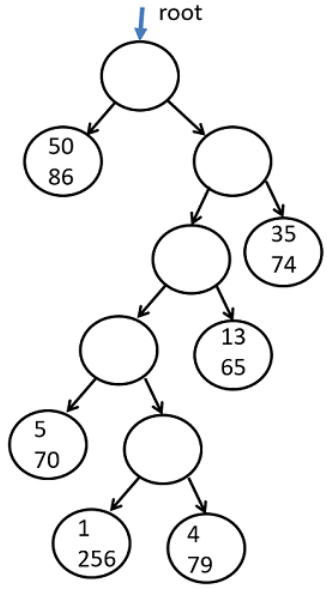
N. The following values are inserted one at time in the order show, left to right, into an initially empty binary search tree using the simple insertion algorithm demonstrated in class. **(NOT the Red - Black tree insertion algorithm)**

What is the depth of the the node that contains the value 5 in the resulting tree?

```
17  30  17  -10  10  12  0  30  5  21
```

_____

O. Consider the Huffman Code tree to the right. In the leaves, the top number is the frequency of the value in the original file and the bottom number is the value from the original file.



Using the Standard Tree Format (STF) from assignment 10, how many bits are needed to encode the tree?

Do not include the 32 bits for the size of the tree that are written to the header before the bits that actually describe the tree, only the bits needed to represent the tree itself.

_____

P. The following method takes 4 seconds to run when n = 1,000,000.

What is the expected time for the method to complete when n = 2,000,000?

Assume the nextInt() method is O(1). _____

```
public static List<Integer> getList1(int n, Random r) {
   LinkedList<Integer> result = new LinkedList<>();
   for (int i = 0; i < n; i++) {
       int x = r.nextInt();

       if (x % 2 == 1)
           result.add(0, x); // position, val to add;
       else
           result.add(x);
   }
   return result;
}
```

Q. Suppose we have an array with 4,000,000 distinct doubles in random order. We are going to perform S searches of the data to determine if various doubles are present in the array. Assume the doubles we search for are never present in the array. We will not use any other data structures.

We can search the array without sorting it OR sort the array, using an efficient sorting algorithm, and then search the array. **For what value of S is the number of times items from the array are accessed *roughly* the same for the two proposed approaches?**

_____

R. The following method takes 20 seconds to complete when n = 1,000,000. **What is the expected time for the method to complete when n = 2,000,000?** The binary search tree class use the simple add method demonstrated in class, implemented iteratively.

_____

```
public static BinarySearchTree<Integer> make(int n) {
    BinarySearchTree<Integer> result;
    result = new BinarySearchTree<>();
    for (int i = 0; i < n; i++) {
        result.add(i * 10);
    }
    return result;
}
```

S. Consider the following initial array. [17, -5, 3, 0, 37 , -5, 14, 12]

The array is being sorted using one of the algorithms demonstrated in class. At some intermediary step during the sorting process, the array contains the following values in the order shown.

[-5, 0, 3, 17, -5 , 12, 14, 37]

Which sorting algorithm demonstrated in class is most likely being used to sort the array?

_____

T. Assume the Strings shown in the table to the right have the hashCode listed.

String	hash code
A	56
D	80
J	116
M	100
S	36
V	50
Z	32

The Strings are inserted into a hash table in the order shown below, left to right. The hash table uses open addressing and linear probing to resolve collisions. The internal array initially has a length of 10. Null elements are shown with a forward slash, /. The load limit is 0.8.

M, A, S, V, D, Z, J

What will the elements of the internal array be after adding the 7 elements? Write out the array in the form [N1, /, /, N2 … /, N7] where / indicates a null element and N1 – N7 are the Strings. Do not include any quotes for the Strings.

_____

U. The following values are added one at a time, in the order shown from left to right, to an initially empty minHeap using the algorithm demonstrated in class.

5 7 5 7 10 12 15 3 -2

What value is in the root of the tree and what is the value in the node with a depth of 3 and second from the left at that level?

_____

V. **Which of the following is the best choice for a hash function for ints being stored in a hash table?** x is an int and h(X) is the proposed hash function. Circle the best answer.

h(x) = x h(x) = x / 1000 h(x) = x % 1000 h(x) = (int) (x * Math.random())

W. **Using the techniques developed in lecture, what is the T(N) (actual number of executable statements given N data elements) of the the following method.** N = data.length

_____

```
// pre: x != 0
public static int T_of_N(int[] data, int x) {
    int r = 0;
    for (int i = 0; i < data.length; i++)
        if (i % 2 == 0) {
            int val = data[i];
            int y = x / (i + 1);
            val *= y;
            r += val;
        } else
            r += data[i] / x;
    return r;
}
```

X. What is output by the following code? _____

```
int x =  IntStream.of(-28, 36, -21, 26,
                -12, 0, 27, -18)
        .filter(n -> n < 0)
        .map(n -> -1 * n)
        .filter(n -> n % 3 == 0)
        .min()
        .getAsInt();
System.out.print(x);
```

Y. A trie, implemented uncompressed as we demonstrated in class, stores the following words. The trie has a root node with no letter. How many nodes, including the root, are in the trie?

_____

dog, dogs, dogsled, doge, dogsleds, cat, cats, catsup

EXTRA CREDIT 1 (1 point): What was your favorite assignment? _____

EXTRA CREDIT 2 (1 point): Two UTCS professors have won the Turing Award. What are their last names?

_____

2. **Trees – 12 points:** Complete a helper method for a tree class that removes all nodes that are initially leaves from the tree that contain a given piece of data.

The tree is a general tree. A node can have any number of children. The Tree and Node classes for this question:

```
public class Tree<E> {
    private Node<E> root; // null i.f.f. size == 0
    private int size; // number of nodes in this tree

    private static class Node {

       private E data; // this is never null

       /* The variable children stores null if this Node a leaf. Otherwise
          all elements of children are non-null if an internal node. */
        private ArrayList<Node<E>> children;
```

The children of a Node are stored in a list, but the order they appear in the list is unimportant. The helper method removes all leaf nodes in the tree that contain a target value. The helper method returns the total number of nodes removed.

```
/* pre: size() >= 2, tgt != null */
public int remove(E tgt) {
    return helper(tgt, root);
}
```

Complete the following helper method.

```
// remove leaves with tgt and return the number of nodes removed
private int helper(E tgt, Node n) {
```

You may not change the method header and on this question AND may NOT add any other helper methods.

Do not create any new data structures.

You may use the methods from the ArrayList class and the Iterator interface (including implicitly with for-each loops).

You may use the equals method on Objects.

Do not use any other Java classes or methods.

Note the precondition that the size of the tree is >= 2, meaning root is not null and the root node has at least a single child. This simplifies the problem somewhat.

```
// remove leaves with tgt and return the number of nodes removed
private int helper(E tgt, Node n) {
```

3. **Huffman Coding – 13 Points: Huffman Coding -** Complete the following private instance method for a HuffmanCodeTree. This question uses a new header format described below. The structure of the tree is based on the new Huffman codes in the same way as presented in class and implemented on assignment 10.

```
/*: pre: bis != null and is positioned before the first bit of the VCF header data,
after the VCF format constant itself. (MAGIN_NUM and VCF already read in.)
Assume the header itself is correct with no errors.
post: returns true if all the codes in the header are present and correct
in this HuffmanCodeTree and all values in TreeNodes associated with said
codes are correct. Returns false if there is any error in the tree
based on the header bis is connected to. */
private boolean verifyTree(BitInputStream bis)
```

On assignment 10 there were two header formats, Standard Count Format and Standard Tree Format. This question uses a new header format VCF, Value and Code Format. The format of this header, after the VCF constant itself is:

[number of codes, expressed as an 8-bit integer]

If the number of codes is N, then the next N parts of the header are as follows: (line breaks added for clarity)

[original value as a 9-bit integer]
[length of the code for the original value as an 8-bit integer]
[the new Huffman Code for the original value]

So, the VCH header format is:

```
[number of codes]
1st of N codes: [original value][length of the new code][new code]
2nd of N codes: [original value][length of the new code][new code]
....
Nth of N codes: [original value][length of the new code][new code]
```

The HuffmanCodeTree has already been constructed. This method is used for debugging purposes to verify all the codes in the header are present and correct in the tree. The method also verifies the value field of TreeNodes are set correctly. The value of internal TreeNodes part of codes in the header must be -1 and the value for leaf nodes for codes in the header are correctly set to the original value from the header for a given code. **The method does not check if there are codes in the tree that are not present in the header. The method does not alter the tree in any way.**

The HuffmanCodeTree class for this question:

```
public class HuffmanCodeTree {
    private TreeNode root;

    private static class TreeNode {
        private int value;
        private int freq;
        private TreeNode left;
        private TreeNode right;
    }
```

You may use the provided TreeNode and HuffmanCodeTree class.

You may use the readBits method from the BitInputStream class:
public int readBits(int howManyBits) Returns the number of bits requested as rightmost bits in returned value.

Do not use any other Java classes or methods. Not even Strings. Do not create any new data structures. Do not use recursion in your answer.

```
private boolean verifyTree(BitInputStream bis) {
```

4. **Graphs – 12 points:** Complete the following instance method for the Graph class from assignment 11.

```
/* pre: requiredHubs > 0, factor > 1.0. post: per the problem description. */
public boolean hasRequiredHubs(int requiredHubs, double factor)
```

The method returns true if the Graph has the required number (or more) of hubs. A hub is a vertex with a degree (number of edges) much higher than the average degree of vertices in the Graph.

For this question a vertex must have greater than or equal to (factor * the average number of edges per vertex in the graph) to be considered a hub.

Recall the Graph and Vertex classes:

```
public class Graph {
    // The vertices in the graph.
    private Map<String, Vertex> vertices;

    private static class Vertex {
        private String name;
        private List<Edge> adjacent;
        private int scratch;
        private Vertex prev; }
```

You may use methods from the Map and ArrayList classes, and the Iterator interface.

Do not create any new data structures.

Do not use any other methods from the Graph or Vertex classes unless you implement them yourself as a part of your answer.

Do not use any other Java classes.

Do not use recursion in your answer.

```
/* pre: requiredHubs > 0, factor > 1.0. post: per the problem description. */
public booolean hasRequiredHubs(int requiredHubs, double factor)
```

5. Linked Lists – 13 points: Complete the **add(String value)** instance method for the LinkedBigInteger class.

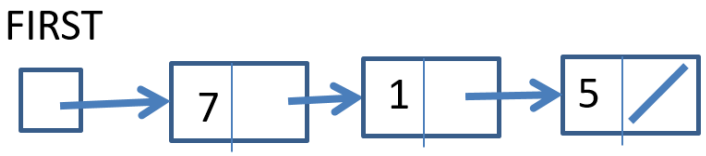
We typically write (and express) integers with the most significant digit as the left most digit:

517 = 5 * 100 + 1 * 10 + 7 * 1

However, we could also write (and express) integers with the least significant digit as the left most digit:

715 = 7 * 1 + 1 * 10 + 5 * 100

The LinkedBigInteger class stores integers with an arbitrarily large number of digits. Each digit is stored as an int in a singly linked node. The least significant digit, the ones digit, is stored in the first node of the structure, the tens digit in the second node and so forth. Here is how the class would store the integer Five Hundred Seventeen.



The LinkedBigInteger class for this question:

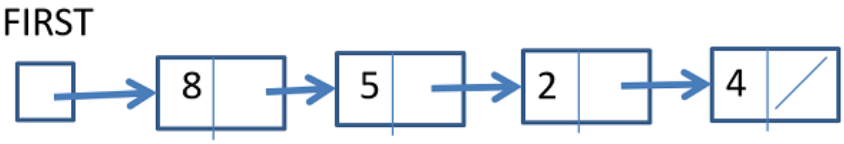
```
public class LinkedBigInteger {
    private final Node FIRST;
    public LinkedBigInteger() { FIRST = new Node(0); }

    private static class Node {
        private int digit;
        private Node next;
        private Node(int d) { digit = d; }
```

Complete the following instance method for the LinkedBigInteger class

```
/* pre: value != null, value.length() >= 1,
    all chars in value are digits, '0' to '9',
    The most significant digits is at index 0. There are no leading 0s. */
    public void add(String value) {
```

Give the LinkedBigInteger above, after the call add("3741") the internal structure would become:

517 + 3741 = 4258
Written with least significant digit first: 8524



Recall you can convert a char from '0' to '9' to the associated int by subtracting the char '0'. For example, int x = '7' - '0'; // x now stores 7.

You may use the String charAt and length methods. You may use the given Node class. Do not use any other Java classes or methods.

Do not create any objects other than new Nodes. **Do not use recursion.**

Do not use any other methods from the LinkedBigInteger unless you implement them yourself as a part of your answer.

```
public void add(String value) {
```