COSC 222 **COSC 222 Final Exam** Tues, June 25, 2019

Student Name: _____

Student ID: _____

Student Signature: _____

This exam has 32 questions, for a total of 84 points.

COSC 222 Final Exam. Summer Term, 2019
Examiner: J. Nastos
Answer the questions in the spaces provided on the question sheets.

RULES GOVERNING FORMAL EXAMINATIONS

1. Each candidate must be prepared to produce, upon request, a UBC Student ID.

2. Students must leave all backpacks and resources at the front or side of the room. All handheld devices (turned off) and notes should be left in backpacks or jackets and located away from your seat. If a bathroom break is absolutely necessary during the exam, students must first ask for permission and empty their pockets of any remaining items before leaving the exam room.

3. Candidates must not destroy or mutilate any examination material, must hand in all examination papers, and must not take any examination material from the examination room without permission of the invigilator.

4. Candidates must follow any additional examination rules or directions communicated by the instructor or invigilator.

1. (1 point) It was said that this course is all about Java Collections. Which of the following is **true** about Java Collections?

   (a) `Set<E>` is an implementing class for `Collection<E>`

   (b) `MultiSet<E>` is an implementing class for `Collection<E>`

   (c) `BinaryTree<E>` is an implementing class for `Collection<E>`

   (d) `HashSet<E>` is an implementing class for `Set<E>`

   (e) `Set<E>` is an implementing class for `HashSet<E>`

2. (1 point) When creating a class and overriding the `.equals()` method, say that I also override the `.hashCode()` method simply with the line `{return 10;}`. Which of the following is correct?

   (a) I will not be able to properly create a HashSet or HashMap of these objects.

   (b) HashSets and HashMaps of these objects will still work like normal.

   (c) HashSets and HashMaps of these objects will still work, but will be slower.

   (d) There will be a compile error because the hash code can't always be just 10.

   (e) I will not be able to compare two objects of this class properly for equality.

3. (1 point) Which of the following most closely describes the Java hash code function used for String objects?

   (a) It is a fixed number every time

   (b) It is the sum of the ASCII values of the characters in the String

   (c) It is the product of the ASCII values of the characters in the String

   (d) It is the value of a polynomial whose coefficients are the ASCII values of the characters in the String

   (e) None of the above.

4. (1 point) Let T be a (regular) trie containing the words {I, drink, coffee, every, day}. If we do not count the empty root node as a node, choose the correct statement:

   (a) T contains five nodes

   (b) T contains five leaves

   (c) The compressed trie of T contains five nodes

   (d) The compact trie of T contains five nodes

   (e) We can search for the substring "every day" in a single root-to-leaf path of T.

Points earned: _____
out of a possible 4 points

5. (1 point) If you perform a DFS on a binary tree starting at its root, and choose left children before right children to break ties, the "start time" (the order in which nodes are placed on the DFS stack) will produce:

   (a) An in-order traversal of the tree

   (b) A pre-order traversal of the tree

   (c) A post-order traversal of the tree

   (d) None of the above

6. (1 point) If you perform a DFS on a binary tree starting at its root, and choose left children before right children to break ties, the "end time" (the order in which nodes are popped off the DFS stack) will produce:

   (a) An in-order traversal of the tree

   (b) A pre-order traversal of the tree

   (c) A post-order traversal of the tree

   (d) None of the above

7. (1 point) If you perform a BFS on a binary tree starting at its root, and choose left children before right children to break ties, this will produce:

   (a) An in-order traversal of the tree

   (b) A pre-order traversal of the tree

   (c) A post-order traversal of the tree

   (d) None of the above

8. (1 point) Say we implement the `SortedSet<E>` interface with a sorted ArrayList by maintaining sorted order after every insert and deletion and discarding duplicates. What are the worstcase runtimes of `.insert(e)` and `.contains(e)` (respectively) if these methods are implemented properly?

   (a) O($\log(n)$) and O(1)

   (b) O($\log(n)$) and O($\log(n)$)

   (c) O(n) and O($\log(n)$)

   (d) O($\log(n)$) and O(n)

   (e) O(n) and O(n)

Points earned: _____
out of a possible 4 points

9. (1 point) Say we instantiate a `SortedSet<E>` with a `TreeSet<E>`. What are the runtimes of `.insert(e)` and `.contains(e)` (respectively)?

   (a) O($\log(n)$) and O(1)

   (b) O($\log(n)$) and O($\log(n)$)

   (c) O(n) and O($\log(n)$)

   (d) O($\log(n)$) and O(n)

   (e) O(n) and O(n)

10. (1 point) Choose the most descriptive and correct statement about **MergeSort**

    (a) It is not stable and runs in $\Theta(n)$ time

    (b) It is stable and runs in $\Theta(n)$ time

    (c) It is not stable and runs in $\Theta(n \log(n))$ time

    (d) It is stable and runs in $\Theta(n \log(n))$ time

11. (1 point) Choose the most descriptive and correct statement about **HeapSort** when the heap is implemented on an array:

    (a) It does not sort in-place and runs in $\Theta(n)$ time

    (b) It is an in-place sort and runs in $\Theta(n)$ time

    (c) It does not sort in-place and runs in $\Theta(n \log(n))$ time

    (d) It is an in-place sort and runs in $\Theta(n \log(n))$ time

12. (1 point) Choose the most descriptive and correct statement about **BucketSort**:

    (a) It does not sort in-place and runs in $\Theta(n)$ time

    (b) It is an in-place sort and runs in $\Theta(n)$ time

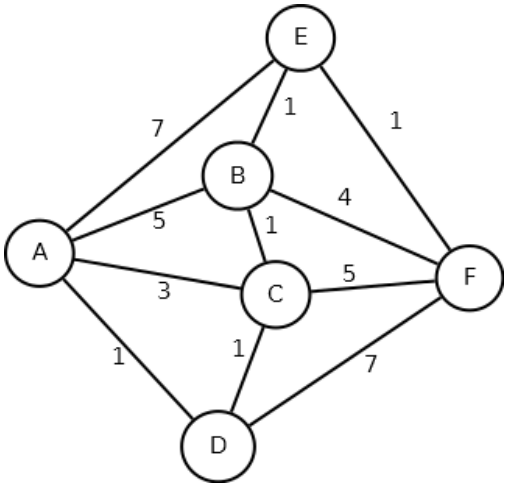
    (c) It does not sort in-place and runs in $\Theta(n \log(n))$ time

    (d) It is an in-place sort and runs in $\Theta(n \log(n))$ time

Points earned: _____
out of a possible 4 points

13. (5 points) The first table below shows the end of the first iteration of running Dijkstra's algorithm on the following graph, starting from vertex A.

    In the empty table that follows it, show the state of Dijkstra's algorithm after **one more iteration**. In the final table (bottom), show the final result after running Dijkstra's algorithm entirely.



	A	B	C	D	E	F
Dist	0	5	3	1	7	∞
Parent	-	A	A	A	A	-
Processed	1	0	0	0	0	0

After one more iteration (2 points):

	A	B	C	D	E	F
Dist	0					
Parent	-					
Processed						

After completion of the algorithm (3 points):

	A	B	C	D	E	F
Dist	0					
Parent	-					
Processed						

Points earned: _____
out of a possible 5 points

14. (5 points) Choose **one** graph concept to use in order to solve each of the problems below. Choose from: BFS, DFS, Eulerian Traversal, Minimum Spanning Tree, Shortest Paths, Topological sort.

    (a) Electric current drops with the length of wire it travels in. A power plant must deliver electricity to many locations (such as homes or towns), and far-away customers will have to pay more to have electricity reach them. The power company wants to set up infrastructure to deliver electricity to these customers so that the on-going cost of electricity is as cheap as possible.

    (b) In a GUI design, we have containers which might contain some content panes and borders and menus, and panes might contain windows and buttons and objects, while menus also contain menu items. The program drawing the GUI needs to draw the lowest-level containers first, and then draw its contents on top of the low-level container, and then the contained objects on top of the content panes, and so on. How can this drawing order be determined?

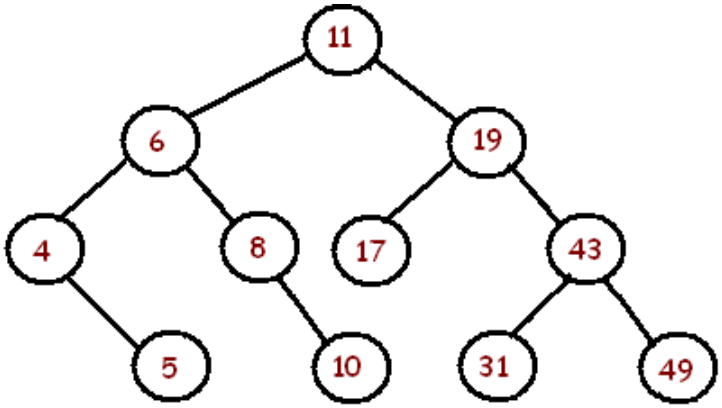
    (c) We are lost in a forest and we have set-up camp at good food source. We need to explore outward in hopes of finding a road to signal to drivers that we need help, but we do not want to get too far from the campsite unless we really have to.

    (d) We are lost in a forest and an unknown scary beast is chasing us and we want to run away until we find a road and can signal drivers for help.

    (e) We have a set of extension cords whose ends are only compatible with certain other extension cords. We want to know if we can connect **all** our cords together so that our lawn mower can reach the end of the yard.

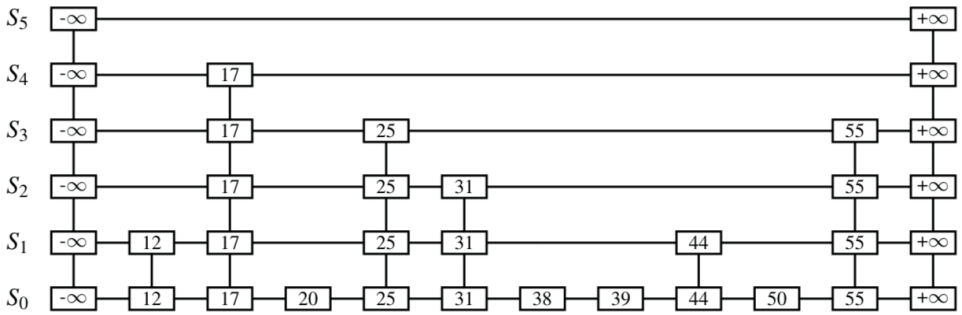
Points earned: _____
out of a possible 5 points

15. (a) (2 points) In the binary search tree below, circle the nodes that could be coloured Red to make it a valid Red-Black tree, if possible. If it is not possible, explain why not.



(b) (2 points) Ignoring the Red-Black colouring from the previous question, treat the tree as a regular binary search tree and draw a resulting tree after a call of `.remove(11)` (you just have to produce one of two possible resulting trees).

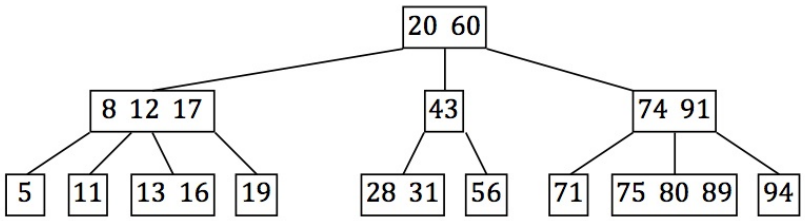
16. In the skiplist below, circle the nodes that will be accessed upon a call of `.contains(40)`.



Points earned: _____
out of a possible 4 points

(a) (2 points) Starting with an initially empty (2,4)-tree, show the (2,4)-tree resulting after inserting the following keys in this order: 2, 3, 5, 7, 11, 13, 17

(b) (2 points) Consider the following (2,4)-tree. Draw the resulting tree after a call of `.remove(60)`.



(c) (2 points) Starting with an initially empty AVL tree, show the resulting AVL-tree after inserting the following keys in this order: 2, 3, 5, 7, 11, 13, 17

Points earned: _____
out of a possible 6 points

17. (a) (2 points) Draw the resulting hash table of size 7 after performing the following set operations (in the order given) using the hash function $h(x) = 4x + 1 \bmod 7$:

    .add(19), .add(15), .add(9), .add(8) .remove(15) .add(1)
    Assume the hash table uses *quadratic probing* as its collision handler.

    (b) (1 point) After the above operations are performed, what is the *load factor* of this hash table?

18. (a) (3 points) Create a trie for the following words: {cat cater cathode code decode decoding} The character index of the start of each word is 0, 4, 10, 18, 23, 30.
    Ensure that your trie stores the information as to where each word exists in this text.

    (b) (2 points) Turn your trie into a compressed trie.

Points earned: _____
out of a possible 8 points

19. (a) (2 points) The following letter frequencies appear in a certain text. Create a huffman tree to encode these letters in a prefix code.

a	b	c	d	e	f	g	h
5	2	4	3	3	1	1	7

(b) (2 points) How many bits would be used to encode text containing these letters with the above frequencies?

(c) (2 points) Since there are 8 distinct letters, we could encode them with a 3-bit fixed-length encoding. Does the prefix code offer any compression over this fixed-width encoding? If so, how many bits are saved?

Points earned: _____
out of a possible 6 points

20. Consider a String pattern `AAATT` over the alphabet {A,T,C,G}. For these questions, adopt the convention that Strings are 0-indexed.

    (a) (1 point) Give the KMP `Failure[]` function for this pattern.

    (b) (3 points) Give the Boyer-Moore `Last[]` function for this pattern, and illustrate this algorithm searching this pattern in the text below by showing the various positions the pattern takes when comparing characters to the text.

A A T T A A T T T T A A T T A A A T T A T A T

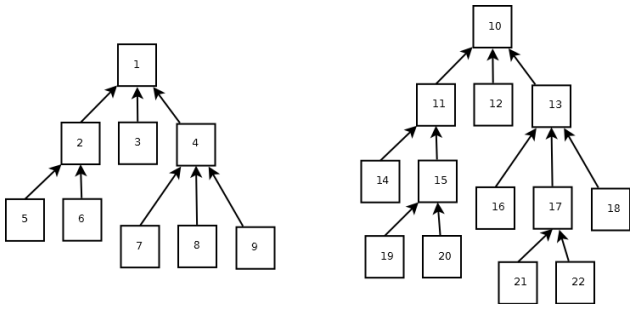
Points earned: _____
out of a possible 4 points

21. Begin with the keys 1, 2, 3, 4, 5, 6, 7, 8, 9 each in its own set.

    (a) (2 points) Merge some of these sets by applying the following sequence of `union` operations and show the resulting Union-Find data structure either in a linked node picture or as an array of parent nodes (your choice). Indicate the relevant height heuristic as well. Break ties by having the larger representative value point to the smaller representative value. Do not use path compression for this part.

    `Union(2,9), Union(1,7), Union(4,6), Union(3,5), Union(2,7), Union(4,5), Union(1,9), Union(2,4), Union(9,7)` .

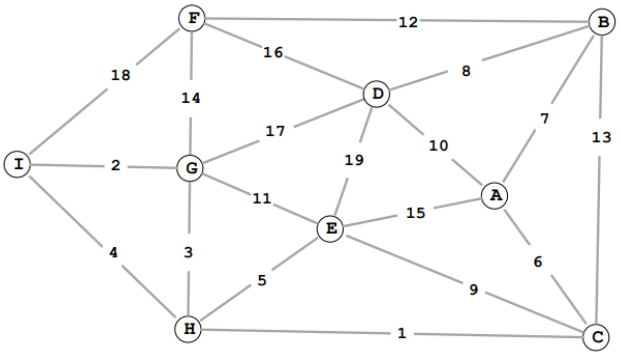
    (b) (2 points) Consider the following Union-Find data structure.

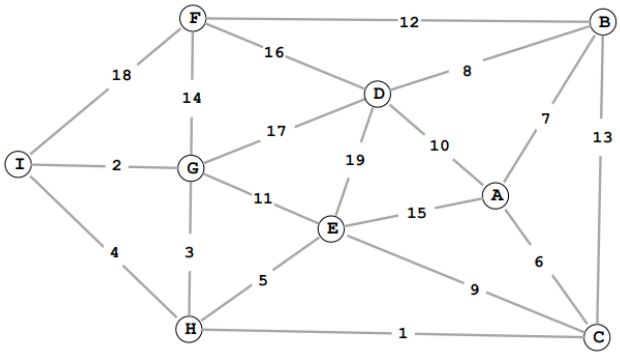


    Assume node 10 has a larger depth rank than node 1. Show the resulting data structure after performing the operation `Union(9,20)` when path compression is included.

Points earned: _____
out of a possible 4 points

22. (4 points) The following graph has 9 vertices and distinct edge weights from 1 to 19. It is duplicated twice so that you can apply different algorithms to each one.





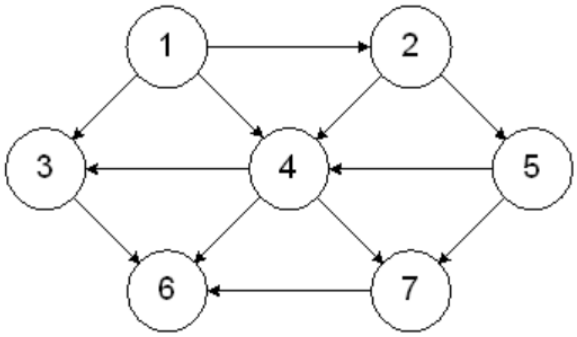
Complete the sequence of edges in the optimal Minimum Spanning Tree in the order that *Kruskal's algorithm* includes them (by writing their edge weights). You do not need to show your work here.

1									

Complete the sequence of edges in the optimal Minimum Spanning Tree in the order that *Prim's algorithm* includes them (by writing their edge weights). You do not need to show your work here. Assume Prim's begins at the source node A.

6									

23. (2 points) Find a valid topological ordering of the nodes in the following directed graph.



Points earned: _____
out of a possible 6 points

24. Write a line (or some lines) of Java code to instantiate a data structure most suited for the following applications.

    (a) (2 points) A `Customer` object contains String name, String city, String address, String phoneNumber, and Integer id. A store wants to verify each customer belongs to their customer loyalty program, and a cashier asks each customer for their phone number to look them up. Because multiple people could have used the same phone number, the cashier will ask the customer if their name is one of the names from a list of names found with that phone number.

    (b) (2 points) A `Car` object has attributes String make, String model, Integer year, String colour. People will search the Car collection for car models between two years (for example, cars from 2008 to 2012). Assume the Car class is not Comparable.

    (c) (2 points) A number of volunteers will watch traffic for a day. Each volunteer can be identified with an integer ID. Each volunteer will count how many cars of each colour he or she sees. We do not know what colours the volunteers may encounter. At the end of the day, each volunteer will have a count for each colour he or she saw. Give one data structure to store all the values from all the volunteers.

25. (2 points) Using one sentence, state what process or algorithm the following recursive code is performing.

```
static boolean zippy(Node n, int t) {
  if (n == null)
    return false;
  if (t == n.data)
    return true;
  if (t > n.data)
    return zippy(n.right, t);
  if (t < n.data)
    return zippy(n.left, t);
}
```


Points earned: _____
out of a possible 8 points

Here is some code for a Graph class from our Lab on Social Networks Recommend-a-Friend. Questions follow on the next page.

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map.Entry;

public class Graph   {

   private HashMap<String, ArrayList<String>> list = new HashMap<>();

   public void add(String s) {
     if (list.containsKey(s) == false)
        list.put(s, new ArrayList<String>());
   }

   public void add(String s, String t) {
      if (list.containsKey(s) == false)
         System.err.println(s + " does not exist ");
      else if (list.containsKey(t) == false)
         System.err.println(t + " does not exist ");
      else if (list.get(s).contains(t) == false) {
         list.get(s).add(t);
         list.get(t).add(s);
      }
   }

   public String toString() {
      String output = "";
      for (Entry<String, ArrayList<String>> e : list.entrySet())
         output += e.getKey() + ": " + e.getValue() + "\n";
      return output;
   }

   public ArrayList<String> getNbrs(String v) {
      return list.get(v);
   }

   public boolean containsName(String s) {
      return list.containsKey(s);
   }
}
```


Points earned: _____
out of a possible 0 points

26. (2 points) **Without** making any edits on the previous page, describe here how you could modify the graph class on the prior page to support edge weights (like in our shortest path or MST instances). Use up to three sentences.

27. (2 points) The graph class on the previous page uses Strings as Nodes. Make edits to the graph class to make the nodes any Generic type. Use a single line to strike-through the relevant code and replace it in a near-by space on the page.

28. (2 points) The given graph class supports simple, undirected graphs. Make a change on the previous page to make the graph class support multiple edges between a pair of nodes. Label these edits with a star or asterisk.

29. (2 points) Describe here, in at most 3 sentences, how you would modify the code so that it would support directed edges.

Points earned: _____
out of a possible 8 points

30. (a) (2 points) Have a look at the following program. Assuming the input sentence fed into the `process()` method has n total words, and d distinct words, estimate the big-O runtime of this method. You do not have to justify your answer.

```
import java.util.ArrayList;
public class BadMap {

   public static void main(String[] args) {
    process("the guy said and and then the guy said guy".split(" "));
   }

   public static void process(String[] inputWords) {
      ArrayList<String> uniqueWords = new ArrayList<>();
      ArrayList<Integer> counts = new ArrayList<>();
      for (String word : inputWords) {
         if (uniqueWords.contains(word)) {
            int position = uniqueWords.indexOf(word);
            counts.set(position, counts.get(position)+1);
         }
         else {
            uniqueWords.add(word);
            counts.add(1);
         }
      }

      for(int i=0; i<uniqueWords.size(); i++) {
         System.out.print(uniqueWords.get(i) + ":" + counts.get(i) + ", ");

      }
   }
}
OUTPUT:
and:2, then:1, the:2, guy:3, said:2,
```

(b) (3 points) The method finds the unique words in a given String array and counts the occurrences of each word. Write a method that does essentially the same as the method above but does so faster (ideally, in O(n) time). Use any Java data structures you know. The output can be formatted as differently as you like.

Points earned: _____
out of a possible 5 points

31. (3 points) Complete the `merge` method below, which is the merge step of MergeSort. That is, it takes two sorted arrays as input and returns a new sorted array that contains all the elements from the two input arrays. Furthermore, it should keep all duplicated values, and it should run in $O(n)$ time, where n is the size of the output array.

```
public static int[] main(int[] a, int[] b) {




}
```

32. (3 points (bonus)) Consider a *full ternary tree* where every internal node gets a left, middle, and right child. We saw that full binary trees can be implemented in an array using formulas `left(i) = 2*i+1, right(i) = 2*i+2, parent(i) = (i-1)/2`.

    If the root of a ternary tree has index 0, figure out expressions to use in order to obtain the index of children and parent nodes in a ternary tree. That is, find the expressions for `left(), middle(), right(), and parent()`

Points earned: _____
out of a possible 3 points