University of Maryland College Park
Department of Computer Science
CMSC132 Spring 2023 Practice Exam

FIRSTNAME, LASTNAME (PRINT IN UPPERCASE):

STUDENT ID (e.g., 123456789):

Instructions

- This exam is a closed-book, closed-notes exam with a duration of 120 minutes and 200 total points.
- You may lose credit if you do not follow the instructions below.
- **At this point, you must write your name and id at the top of this page and add your directory id (e.g., terps) at the end of odd-numbered pages.**
- Please use a pencil to answer the exam.
- **Do not remove the exam's staple, and do not bend any of the pages, as doing so will interfere with the scanning process.**
- Provide answers in the rectangular areas. If you continue a problem on another page(s), make a note. **For multiple-choice questions, please fill in the bubble (do not circle).**
- **Your code must be efficient and as short as possible.**
- You don't need to use meaningful variable names; however, we expect good indentation.
- You must stop writing once the time is up.

Grader Use Only

I am not sharing the table of all of the topics, but I have still grouped them in an order below…

[redacted]

Problem #1

1. Write your own Thread class that prints the numbers 1 to 20 to console:

2. Write the code in a driver class to initialize an ArrayList of 5 threads, have execute and terminate then print "DONE!" as the last line in the console:

3. What is a deadlock?

4. When should you use a synchronized method?

5. Write code that has a data race:

6. Now, using a synchronized block, solve your data race:

Problem #2

1. Write an abstract class called Shapes. What is an abstract class you can write? Why is it a good candidate for an abstract method?

2. What is dynamic binding?

3. A method called doSomething() throws a ClassCastException. Write the code to handle said exception:

```
doSomething();
```

4. The **Car** class extends the **Transportation** class. Which of the following are valid (will compile)? Fill in the bubbles of the valid ones.

(a) ArrayList<Transportation> a = new ArrayList<Car>();
(b) ArrayList<? extends Animal> b = new ArrayList<Car>();
(c) ArrayList<?> c = new ArrayList<Transportation>();

5. Create a heap from the following Array representation:

1	3	6	5	9	8
0	1	2	3	4	5

6. What type of heap is this? Then, daw the heap (as a tree) that would result from deleting **1** from the following heap.

Problem #3

Given the following implementation of a binary search tree.

```
public class BinarySearchTree {
    //Represent the node of binary tree
    public static class Node{
        int data;
        Node left;
        Node right;
        public Node(int data){
            //Assign data to the new node, set left and right children to null
            this.data = data;
            this.left = null;
            this.right = null;
            }
        }
```

1. Write the **recursive method** method toArray which takes the root of a binary search tree and returns the binary search tree in an array ***in order***.

```
public int[] toArray(Node root){
```

2. Write the recursive method, findUnder, which will return a count of all Nodes that are less than the parameter **upper.**

```
public int findUnder(Node root, int upper){
```

Problem #4

a. (12 pts) What is the big O of the following code? Why do the critical sections of code lead you to believe that?

a.

```
for (i = 1; i <= n+10; i++) {
   for (k = 1; k <= n*10; k += 2) {
      System.out.println(i * k);
   }
}
```

b.

```
for (k = 1; k <= n; k *= 4) {
      for (m = n; m>0; m--) {
         System.out.println(m);
      }
}
```

Problem #4

1. What is the time complexity (big O) of:

a. Merge Sort
b. Bubble Sort
c. Quick Sort

2. What is a stable algorithm? What is in place?

3. Perform the first iteration of insertion and quick sort on the following set of numbers: 7, 13, 12, 9, 20, 16
   b. ***By first iteration, for insertion sort that means selecting which numbers you are inserting and where, for quick sort it means finding the pivot and doing the first set of operations before splitting the array.***

Problem #5

1. Insert the following values into a hash table of size 5 using the hash equation (4x +1) % 5 using the quadratic probing technique, ***with a load factor of .5.*** Hint:use the following function to handle collisions: f(n) = hash(val) + i^2.
   Insert these values in sequential order: 2, 5, 9, 3 15, 9

2. Give an example hash function and table size that would not be effective. Why? What makes a hash function inefficient/ineffective?

Problem #6

1. If I have a rule for which my algorithm can never violate when trying to find a solution, what type of algorithmic strategy might I be using?

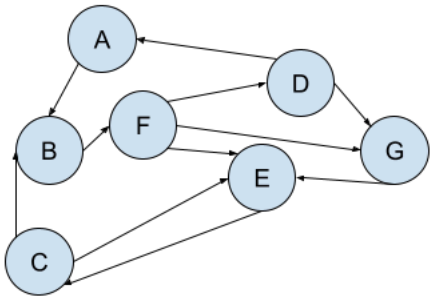
2. How does a branch and bound algorithm work?

3. What are some examples of backtracking algorithms that we have used in class?

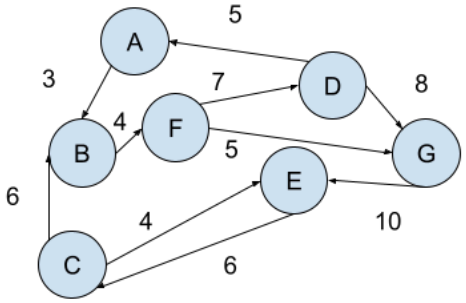
4. What is a greedy approach to the traveling salesman problem?

Problem #7

1. Do a breadth first and depth first traversal of the following graph:



2. (16 pts) Single source shortest paths



Apply Dijkstra's algorithm using **B** as the starting (source) node. Indicate the cost and predecessor for each node in the graph after processing two nodes (**A** and another node). Remember to update the appropriate table entries after processing a node (after it has been added to the set of processed nodes). An empty table entry implies an infinite cost or no predecessor. **Note: you will receive no credit if you simply fill in the entire table with the final costs and predecessors instead of showing the table in the first two steps.**

After processing the first node. Write the name of the node you are processing here: ____**B**___

Node	A	B	C	D
Cost				
Predecessor				

After processing the second node. Write the name of the node you are processing here: _____

Node	A	B	C	D
Cost				
Predecessor				

After processing the Third node. Write the name of the node you are processing here: _____

Node	A	B	C	D
Cost				
Predecessor				

After processing the Fourth node. Write the name of the node you are processing here: _____

Node	A	B	C	D
Cost				

Predecessor				

Problem #8

1. To start, write the code for a Node for a linked list that will store:
   - next
   - price
   - nameOfProduct

```
public class LinkedList{

    private class Node{
      //begin here
```

2. Then, write the implementation for findAndReplace, which is a method that will find a Node with the same nameOfProduct as the parameter **itemToFind**. Then it will replace the Node with a new Node given through the parameters. Note: ***This should be a non-recursive function and you may not just change the price on an existing node, you should replace the entire node.***

```
public void findAndReplace(String itemToFind, Node newNode){
```