COP 3503 Spring 2024 Section 1 Exam #2

Sheet 1: Algorithm Analysis, Extra Sorts

Last Name: ____________________ , First Name: ______________________

Recitation Time (Circle One): 8:30 9:30 10:30 11:30

1) (6 pts) You've run an algorithm that processes an array of size ***n***, for several values of ***n*** and have recorded the execution times in the table below. Use the technique shown in class (blank columns have been provided for your use) and determine the most likely Big-Oh run time of the algorithm in terns of ***n***.

n	***Run time (ms)***				
10000	98				
50000	1162				
100000	3016				
200000	9231				

____

2) (8 pts) Show the result of each iteration of a Radix Sort on the following values. The last column has been filled in.

Initial List	First Sort	Second Sort	Third Sort	Fourth Sort	Sorted List
32361					24169
48147					32361
32661					32647
58347					32661
34629					34629
24169					48147
48341					48341
32647					58347

3) (11 pts) Let T(n) represent the average number of nodes with two children in a binary search tree. Assuming that the root node of a binary search tree has probability of $\frac{1}{n}$ of being any particular rank (from 1 to n) in the sorted list of values stored in the tree, write down a recurrence relation that T(n) satisfies **and simplify this recurrence so that there's a single sum added to a single term on its right-hand side**.

COP 3503 Spring 2024 Section 1 Exam #2

Sheet 2: Greedy Algorithms

Last Name: ____________________ , First Name: ______________________

Recitation Time (Circle One): 8:30 9:30 10:30 11:30

4) (10 pts) A Huffman tree has been created for a particular input file comprised solely of the characters ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’ with the codes listed:

Letter	**Huffman Code**	**Possible Letter Frequency**
‘a’	1111	
‘b’	10	
‘c’	11101	
‘d’	0	
‘e’	111001	
‘f’	110	
‘g’	111000	

(a) Draw the corresponding Huffman Tree below.

(b) Fill in possible letter frequencies, ***without any tie cases***, for each of the seven letters in both the tree and the table above that could create this set of Huffman codes. (Note: Both the frequencies of each letter must be distinct AND when comparing items in the PQ, there should never be a tie between the second and third smallest items in it.

5) (15 pts) Arup's World Wide Coffee Emporium sells cups of coffee. He sells several different types of coffee. Each type of coffee has the following attributes: cost (# of cents it costs Arup to make one cup of this type of coffee), sell (# of cents Arup sell's each cup of this type of coffee), and numcups (the number of cups he has in inventory of this type of coffee.) For this question: write the compareTo method for the coffeecup class AND write a function which takes in an array of type coffeecup storing Arup's total inventory and an integer, totalcups, and returns the maximum profit he can make by selling exactly totalcups number of cups of coffee.

```
class coffeecup implements Comparable<coffeecup> {
     public int cost;
     public int sell;
     public int numcups;

     // Constructor omitted.

     public int compareTo(coffeecup other) {







     }
}
public static long maxProfit(coffeecup[] list, int totalcups) {

















}
```

COP 3503 Spring 2024 Section 1 Exam #2

Sheet 3: Unweighted Graph Algorithms

Last Name: ___________________ , First Name: _____________________

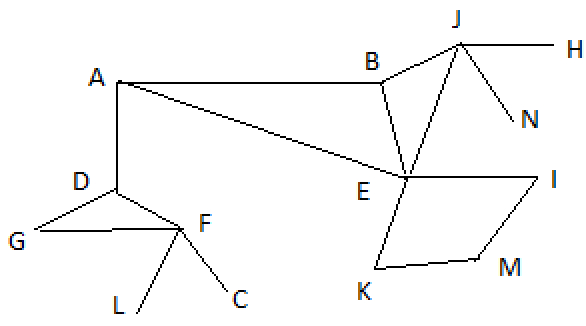
Recitation Time (Circle One): 8:30 9:30 10:30 11:30

6) (10 pts) For this problem, you must figure out the lexicographical first ordering of completing tasks 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, given the following constraints. Each constraint is given in the form (a, b) indicating that task a must be completed before task b.

(5, 2), (3, 1), (7, 4), (4, 3), (10, 2), (6, 3), (8, 10), (9, 5)

_____ , _____ , _____ , _____ , _____ , _____ , _____ , _____ , _____ , _____

7) (7 pts) Show the order in which the vertices in the following graph get visited in a **BFS** starting at vertex **E**. Whenever looping through the neighbors of a vertex, always go through those in alphabetical order.



___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

8) (8 pts) Imagine a building with ***n*** floors labeled 0 through ***n*** – 1, an elevator which can either move you ***a*** floors up or ***b*** floors down, and that you start on floor, ***s*** ($0 \le s < n$). You can use a depth first search to mark all of the floors that are reachable. Complete the recursive dfs method below so that it fills in the boolean array, used, so that after the initial depth first search completes, used[i] will be set to true if and only if it's possible to reach floor i from floor ***s*** via a series of elevator moves, each of which go a floors up or b floors down. (Note: a and b are both guaranteed to be positive integers.)

```
public static void dfswrapper(int n, int a, int b, int s) {
    boolean[] used = new boolean[n];
    dfs(n,a,b,s,used);
}

// Marks all reachable floors in a building of n floors starting
// at the floor cur with an elevator that can go up a floors or
// go down b floors.
public static void dfs(int n, int a, int b, int cur, boolean[]
used) {









}
```

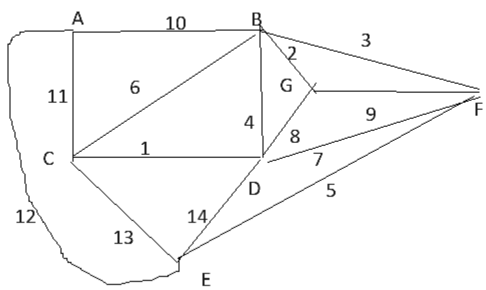
COP 3503 Spring 2024 Section 1 Exam #2

Sheet 4: weighted Graph Algorithms

Last Name: ________________________ , First Name: __________________________

Recitation Time (Circle One): 8:30 9:30 10:30 11:30

9) (10 pts) Trace through Kruskal's Algorithm on the graph drawn below. Show, in order, each edge that is considered to be added to the minimum spanning tree and then put either "yes" or "no" in the second column to indicate whether or not that edge actually gets added to the minimum spanning tree.



Edge to Consider	**Is it Added? (yes/no)**

Note: More rows than necessary are provided. For full credit, you must stop filling out rows as soon as the minimum spanning tree is complete.

10) (10 pts) Let **G** be a directed graph with **n** vertices, where **n** < 1000. Each edge in **G** is assigned a color: red, green or blue. We are given a start vertex in the graph, **s**, and a destination vertex in the graph, **e**. Devise an algorithm (describe in words) to find a path from **s** to **e**, which minimizes the number of red edges on the path. If two paths contain the same number of red edges, the number of green edges should be minimized. If two paths contain the same number of red and green edges, then the number of blue edges should be minimized. The goal of the algorithm is to simply output the number of red, green and blue edges on a path that achieves this objective. Be as clear in your description as possible. You may state the use of any of the algorithms taught in class, but must clearly and unambiguously state the input you're using to that algorithm. (Hint: Any shortest path in a graph of **n** vertices with positive edge weights only can not contain more than **n** – 1 edges.)

11) (5 pts) March 14th is often called Pi Day because the irrational value of Pi, rounded to 2 decimal places is 3.14. Those who are not mathematically inclined often use this day to celebrate a common dessert. What is that dessert?

____