ECE368 Final Exam

Spring 2016

Saturday May 7, 2016.

8:00-10:00 am

FRNY G140

READ THIS BEFORE YOU BEGIN

This is a *closed-book, closed-notes* exam. Electronic devices are not allowed. The time allotted for this exam is exactly 120 minutes.

Always show as much of your work as practical - partial credit is largely a function of the *clarity and quality* of the work shown. *Be concise*. It is fine to use the blank page opposite each equation (or at the back of each question) for your work. Do draw an arrow to indicate that if you do so.

This exam consists of 10 pages; please check to make sure that all of these pages are present before you begin. Credit will not be awarded for pages that are missing – it is *your responsibility* to make sure that you have a complete copy of the exam.

IMPORTANT: Write your login at the TOP of EACH page. Also, be sure to *read* and *sign* the *Academic Honesty Statement* that follows:

"In signing this statement, I hereby certify that the work on this exam is my own and that I have not copied the work of any other student while completing it. I understand that, if I fail to honor this agreement, I will receive a score of ZERO for this exam and will be subject to possible disciplinary action."

Printed Name:

Login:

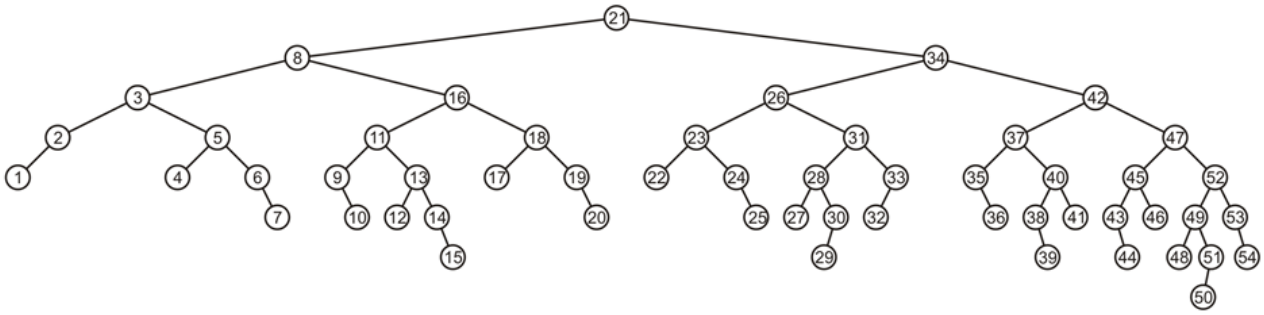
Signature:

DO NOT BEGIN UNTIL INSTRUCTED TO DO SO …

1. AVL Trees (20 Pts Total)

a) Describe the algorithm for inserting a node in an AVL tree. You can write your answer in code (C or C++) or pseudo code listed in bullet points (make sure to account for all the details if you do so).

b) Show the arranging of the tree after the following operation: Delete 1.

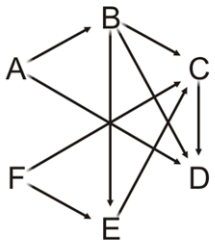


2. Graphs (25 Pts Total)

a) Does every Directed Acyclic Graph (DAG) has a vertex with in-degree = 0? Circle either true or false and prove your answer.

True **False**

b) **Topological sort and Critical Path:**
Suppose we are performing topological sort on the following graph and we are given the Task Time for each node. Fill the values for the **Queue** and update the values for the columns: ***In-Degree***, ***Critical Time***, *and* ***Previous Task***, for the first 4 steps. Whenever you have to pick nodes with the same degree, enqueue in reverse alphabetical order.
(For example, in the first step, you would enqueue ***F*** first, and then ***A***).

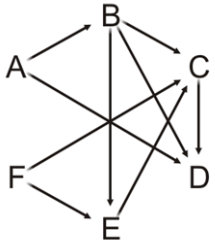


Queue

A	**F**		

Task	In-degree	Task Time	Critical Time	Previous Task
A	0	5.2	0.0	Ø
B	1	6.1	0.0	Ø
C	3	4.7	0.0	Ø
D	3	8.1	0.0	Ø
E	2	9.5	0.0	Ø
F	0	17.1	0.0	Ø

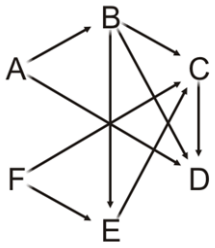
< Step 1 >



Queue

Task	In-degree	Task Time	Critical Time	Previous Task
A		5.2		
B		6.1		
C		4.7		
D		8.1		
E		9.5		
F		17.1		

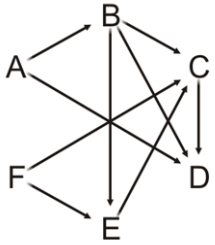
< Step 2 >



Queue

Task	In-degree	Task Time	Critical Time	Previous Task
A		5.2		
B		6.1		
C		4.7		
D		8.1		
E		9.5		
F		17.1		

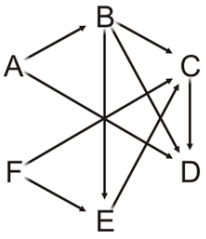
< Step 3 >



Queue

Task	In-degree	Task Time	Critical Time	Previous Task
A		5.2		
B		6.1		
C		4.7		
D		8.1		
E		9.5		
F		17.1		

< Step 4 >



Queue

Task	In-degree	Task Time	Critical Time	Previous Task
A		5.2		
B		6.1		
C		4.7		
D		8.1		
E		9.5		
F		17.1		

3. Sorting (20 Pts Total)

a) List the average case asymptotic run time complexity of the following algorithms (you may list any needed assumptions):

- Merge sort
- Quick sort
- Insertion sort
- Heap sort
- Bucket sort

b) What is the best case scenario for heap sort? Justify your answer.

c) What is the maximum number of inversions that can be removed with one comparison in merge sort? Justify your answer.

4. Class Participation (10 Pts Total)

a) You are given a Doubly-Linked List with one pointer of each node pointing to the next node just like in a Singly-Linked list. The second pointer however CAN point to ANY arbitrary node in the list, not just the previous node. Now write a program/algorithm in O(n) time to duplicate this list.

5. Search Problems (25 Pts Total)

a) Explain the following distances for the N puzzle problem in brief:

- The discrete distance

- The Hamming distance

- The Manhattan distance

b) When using A* search algorithm, choose the best heuristic for the N puzzle problem from part a) and explain the reason.

c) Which data structure should be used to store a 1 billion node search tree on a hard drive? Justify your answer

d) Which of the following algorithms is the fastest to solve the All-pairs Shortest Path problem? Does the answer depend on the graph? If yes, explain how?

- Applying Dijkstra's algorithm for each node

- Floyd-Warshall Algorithm

6. Extra Credit Question (10 Pts Total)

a) No connected graph is a forest. Circle either true or false and prove your answer.

True **False**

Question	Score
Q1	/20
Q2	/25
Q3	/20
Q4	/10
Q5	/25
Extra	/10
Total	/100