CS210 Data Structures (172) Final Exam

Name:_____ ID_____

Instructions:

- This exam contains four questions with multiple parts.
- Time allowed: 180 minutes
- Closed Book, Closed Notes.
- There are 10 pages in this exam booklet.
- **Use of Calculators and / or computing devices / smartphones etc is strictly prohibited.**
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- When the "time is over" is called, it is students responsibility to submit his exam to the invigilator. Submitting completed exam 3 minutes after the "time is over" will incur a penalty of 5 points.

Few gentle reminders:

- If you get stuck on some problem for a long time, move on to the next one.
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You should be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem.
- Keep the points distribution in mind when deciding how much time to spend on each problem.

Do Not write below this line:

a	b	a	b	a	b	a	b	Total:
c		c	d	c		c		

CLO Assessment

CLO1 – Q1(10)	CLO2 – Q2 (10)	CLO3 – Q4 (10)	CLO4 – Q3a (4)

Q1 (a) [2]. Give the Big-Oh notation for the following functions

$f(N) = N^2 \log N^2$	
$f(N) = (N^3 . (5 + N^2))^3$	
$f(N) = \log N^2 + \log \log N^2$	
$f(N) = 1000 \log \log N + 1000. \log N$	

Q1 (b) [4]. Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable ***n***.

`public int module(int n) {`
`  if (n == 0)`
`  return 0;`
`  else if (n == 1)`
`  return 1;`
`  else`
`  return module(n - 1) + module(n - 2);`
`}`	

`public void module2(int n) {`
`  int temp = 0;`
`  for (int i = 0; i < n; i++) {`
`  int j = 0;`
`  while (j < n) {`
`  temp++;`
`  j = j / 2;`
`  }`
`  }`
`}`	

Q1 (c) [4]. Describe the most time-efficient way to implement the operations listed below. Give the run-time for your procedure in Big-Oh notation.

A method that adds (sum) all values in a binary tree. Assume an Integer binary tree stored in an integer array is passed as a parameter.	
Given an adjacency matrix for a digraph (directed graph); count the out-degree for all vertices. (given in the order of V and/or E).	
Assume we change the rules for AVL tree; all larger values go left; smaller values go right to a parent node. Remove a node from this AVL tree.	
Number of swaps using insertion sort for an array which is already sorted.	

Q2 (a) [2].

What is the minimum and maximum number of nodes in an AVL tree with height **6**?

Min:

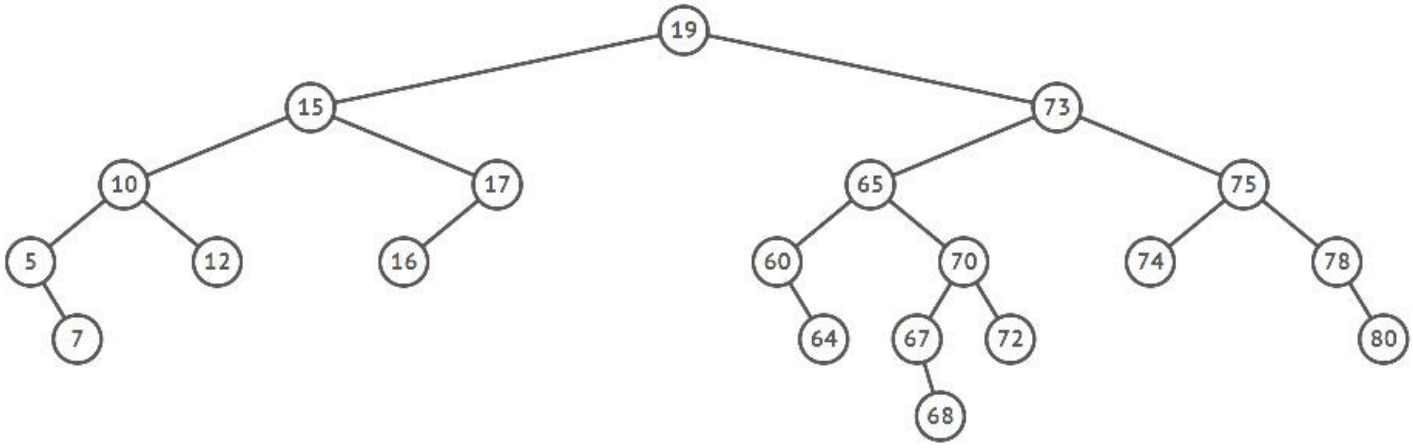
Max:

Let's say the height of a binary tree that stores strings, is **h**. What is the maximum number of internal nodes in this tree? Hint. Internal nodes are non-leaf nodes.

Q2 (b) [3] Illustrate a step-by-step result of inserting the following values in a binary max-heap.

K, D, C, G, H, A, J, B.

Q2 (c) [3] Remove the following keys from the Binary Search Tree given below. Show your tree after each deletion; identify the cases that apply.



19, 17, 15, 80, 64

Q2 (d) [2]. This relates to the resulting BST in the previous part:

How many comparison operations were performed when deleting 19?	
What would be the Big Oh complexity if you are asked to convert this BST to an AVL tree?	

Q3 (a) [4]. Draw the contents of the two hash tables below after inserting the values shown. Show your work for partial credit. If an insertion fails, please indicate which values fail and attempt to insert any remaining values. Use linear probing for collision resolution.

H(k) = k mod array_size

Insert: 12, 4, 18, 6, 14, 21, 9

0	
1	
2	
3	
4	
5	
6	

How many probes?

_____

0	
1	
2	
3	
4	
5	
6	
7	
8	

How many probes?

_____

Q3 (b) [2]: A Prince Sultan University, an undergraduate CS major student needs to successfully complete certain courses to graduate. The Registration office has given you a adjacency list. Draw a digraph.

CS101	CS102, CS175
CS102	CS210
CS175	CS285
CS210	CS330, CS340, CS331, CS311
CS285	CS330, CS311
CS330	CS409
CS331	CS409
CS311	CS492
CS409	CS492
MATH111	CS285
ETHC303	CS492

Q3(c) [4]: Provide a Topological sort starting at CS101. What should be the order of taking courses? Show your work for partial credit.

Q4 (a) [4]: Apply Quick-sort to the following data. Assume, this data is already shuffled. Identify pivot values and show all steps, as a single processor computer would process this data.

mid
low
law
pie
big
paw
ate
bad
sad
ape

Q4 (b) [4]. Give Big-Oh notation for the following problems. For partial credit explain your algorithm.

Given an array of integers of size N with randomly arranged data, if we use selection-sort to sort integer data in ascending order, what would be the run-time?	
Given an array of integer values of size 2*N where all the data is already sorted in ascending order. Give most efficient way of reversing the contents of this array so that the items in array would be sorted in descending order.	
What will be the run time of sorting an integer array with size N if we use binary heap tree for it?	
Which is the most space efficient method for storing a digraph?	

Q4(c) [2]: Suppose you are hired as an intern at a financial organization called Acme Inc. that holds records of their clients' accounts and bank transactions. Acme Inc has more than 4 million account holders, on average there are more than 50,000 transactions taking place every 24 hours. At the end of a month, Acme Inc produces a report listing all the clients ordered by the number of transactions for each client per month.

Your IT manager wrote a program that reads the live data from the servers and stores in a file. His program reads this file line by line to identify an account number and count the number of transaction for each account. There are two problems.

1. The current process is too slow.
2. It takes a few minutes to download the entire data. While he is downloading the transactions data he surely missed some transactions. It is important to have all transactions appear in the report.

Since you have the technical ability of writing efficient algorithms, what would you advise him to do. Be specific about the data structures you would use and provide a run time for your suggestion.