King Fahd University of Petroleum & Minerals

College of Computer Science and Engineering

Information and Computer Science Department

First Semester 131 (2013/2014)

ICS 202 – Data Structures

Major Exam 2

Sunday, November 24th, 2013

Time: 120 minutes

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID# |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section 01 |  | Question # | Max Marks | Marks Obtained |
| Dr. Ramadan |  | 1 | 20 |  |
| 10-10:50am |  | 2 | 20 |  |
| Section 02 |  | 3 | 10 |  |
| Dr. Sami |  | 4 | 10 |  |
|  |  | 5 | 20 |  |
|  |  | 6 | 20 |  |
| 09-09:50am |  | Total | 100 |  |

**Instructions**

1. **Write your name and ID in the respective boxes above and circle your section.**
2. **This exam consists of 9 pages, including this page, plus one reference sheet, containing 6 questions.**
3. **You have to answer all 6 questions.**
4. **The exam is closed book and closed notes. No calculators or any helping aids are allowed.**
5. **Make sure you turn off your mobile phone and keep it in your pocket if you have one.**
6. **The questions are not equally weighed.**
7. **The maximum number of points for this exam is 100.**
8. **You have exactly 120 minutes to finish the exam.**
9. **Make sure your answers are readable.**
10. **If there is no space on the front of the page, feel free to use the back of the page. Make sure you indicate this in order not to miss grading it.**

**Q.1: (20 points: 2x10):** Consider the following tree:



1. List the nodes of the left subtree of the root:
2. List the ancestors of node 15:
3. List the proper ancestors of node 15:
4. List the proper descendents of node 30:
5. What is the degrees of nodes 20, 15, and 32 respectively?
6. List all internal nodes of the above tree:
7. Are nodes 15 and 25 siblings?
8. What are the levels of nodes 7, 15, and 20 respectively?
9. What are the heights of nodes 30, 25, and 32 respectively?
10. Is the above tree full or complete?

**Q.2: (20 points)** AVL Trees

* 1. **(15 points)** Insert the following nodes into an empty AVL tree. Make sure you show each step, including rotations if any.

27, 28, 37, 10, 40, 35, 25, 60

* 1. **(5 points)** Consider the following AVL Tree

Delete the node with key 15 and show the resulting AVL tree. Make sure you show each step, including rotations, if any.

**Q.3 (10 points)** Consider the following definition of the Binary Tree:

public class BTNode<T extends Comparable<? super T>> {

 protected T el;

 protected BTNode<T> left, right;

 public BTNode() {

 left = right = null;

 …

 }

}

public class BT<T extends Comparable<? super T>> {

 protected BTNode<T> root = null;

 public BT() {

 }

 …..

}

Write a method called ***isHeap*** in class BT to determine whether a given binary tree is a heap (max or min). The method should return a boolean and in case it is a heap, prints a message telling if it is a max-heap or min-heap.

**Q.4 (10 points):** Consider the following summarized version of Algorithm mergeSort:

1. static void mergeSort(int[] A) {
2. if (A.length > 1) {
3. int q = A.length/2;
4. int[] leftArray = Arrays.copyOfRange(A, 0, q-1);
5. int[] rightArray = Arrays.copyOfRange(A, q, A.length-1);
6. mergeSort(leftArray);
7. mergeSort(rightArray);
8. A = merge(leftArray,rightArray);
9. }
10. }

Assuming that the merge function in the last line of the algorithm will carry out $n/2$ element comparisons in the best case, and *n* – 1 element comparisons in the worst case, when Algorithm mergeSort is called to sort *n* elements. For simplicity, assume that *n* is a power of 2.

1. (4 points) Derive the recurrence equation describing the **best case** time complexity of the algorithm in terms of the number of element comparisons.
2. **(**6 points) Solve the recurrence equation in part 1, and express it in terms of Big O() notation.

**Q.5 (20 points):** Consider the following directed graph:



Answer the following questions:

**Note:** In the traversals, if at any point there is more than one possible vertex to visit, visit them in the order of their labels.

* 1. (5 points) List the vertices in the order they will be visited using pre-order depth first traversal, starting from vertex v1, draw the depth-first traversal tree
	2. (5 points) List the vertices in the order they will be visited using post-order depth-first traversal, starting from vertex v1 and draw the depth-first traversal tree
	3. (5 points) List the vertices in the order they will be visited using breadth-first traversal, starting from vertex v1 and draw the breadth-first traversal tree
	4. (5 points) List the vertices in the order they will be visited using Topological order traversal. Draw the resulting graph.

**Q.6 (20 points):** Heap:

1. (10 points) Construct a minHeap by enqueuing the following values, respectively. Show the heap after each enqueue operation.

**30, 40, 24, 16, 35, 27, 60, 18, 29, 28, 25.**

1. (10 points) Draw the minHeap after performing each one of the following operations on the resulting minheap in “1”.

**Note:** The operation in “b” must be applied on the resulting minheap from part “a”, and the operation in “c” must be applied on the resulting minheap from part “b”. Show all your intermediate steps.

* 1. (3 points) Delete(60).
	2. (3 points) Delete(16).
	3. (4 points) Change the value of 18 by 70.

**Quick Reference Sheet**

 ,  , , 

