

Name: _____

Roll number: _____

Section: _____

FAST – National University of Computer and Emerging Sciences, Lahore
Data Structures and Algorithms CS210 – Spring 2009
Midterm 2

Time allowed: 90 minutes

Date: 18-Apr-2009

Total Marks: 50

-
- **Make sure this booklet contains 9 pages. (8 printed pages + 1 sheet for rough work)**
 - **Write your roll number on each page.**
 - **You are not allowed to ask any questions.**
 - **Write neat and well commented programs. Seemingly encrypted and convoluted answers will not be decrypted.**
 - **The course has been taught using Java and C++, as a consequence the programming answers may be written in either languages.**
 - **Use ONLY the space provided for your answers. YOUR ANSWER WILL NOT BE GRADED IF WRITTEN ANYWHERE ELSE**
 - **This is a closed book exam**
-

Question #	Total Marks	Marks Obtained
1	10	
2	10	
3	10	
4	10	
5	10	
Total	50	

Good Luck!

Name: _____

Roll number: _____

Question 1: [Marks: 10]

Write a function MergeLists() which takes two sorted (in ascending order) integer singly linked lists and merges them to produce one sorted (in ascending order) singly linked list in time complexity $O(n)$ where n is the total number of elements in the two singly linked lists. Moreover, you are not allowed to declare any temporary list.

Node MergeLists(Node listA, Node listB)

// where listA and listB are the first nodes of the two sorted linked lists

{

}

Name: _____

Roll number: _____

Question 2: [Marks: 10]

Show how the basic version of Heap Sort, that we saw in class, would sort the following sequence of numbers in ascending order (show your steps using trees): 16 3 18 9 8 1 15 22 11

Name: _____

Roll number: _____

Question 2:

Name: _____

Roll number: _____

Question 3: [Marks: 10]

Show how the basic Quicksort algorithm that we saw in class, would sort (in ascending order) the following array of numbers:

17 8 19 2 15 5 16 4 14 13 6 10

Show each pass (you need not show the intermediate steps involved) of the algorithm clearly, with the pivot element highlighted.

Name: _____

Roll number: _____

Question 4: [Marks: 10]

Write a simple recursive algorithm that would take an input number, NUM, and then return all items that are bigger than or equal to this number in a given Binary Search Tree. Make sure that your algorithm runs in $O(n)$ time in the worst case where n is the total number of nodes in the tree and takes advantage of the properties of Binary Search Trees.

```
void LargerOrEqual(int NUM, BSTNode h)
{
```

```
}
```

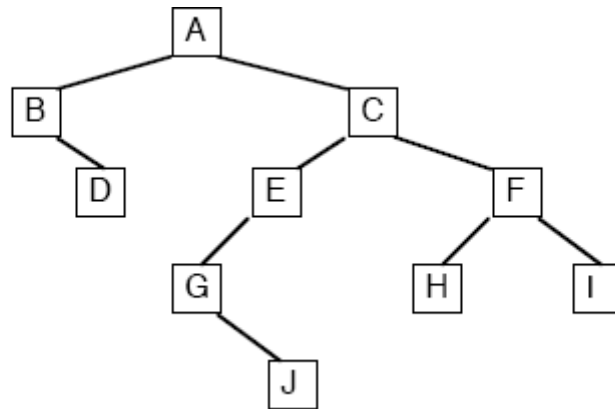
Name: _____

Roll number: _____

Question 5 [Marks: 10]:

a) Marks [3]

Given the following binary tree



List the nodes in the following orders:

Inorder : _____

Preorder : _____

Postorder : _____

b) Marks [3]

Please specify the properties mentioned in the following table for the basic version of each algorithm that is listed :

Algorithm	Worst case time complexity	Stable? (Yes/No)	In-Place? (Yes/No)
Bubble Sort			
Quick Sort			
Merge Sort			
Heap Sort			

Name: _____

Roll number: _____

Question 5 :

c) Marks [4]

The following array of numbers represents a MaxHeap. Draw this heap and then show the resultant heap after the Delete operation is carried out on it.

30	19	8	17	15	6
----	----	---	----	----	---

Name: _____

Roll number: _____

ROUGH WORK