Department of Computer Science

CS 201 – Data Structures

Final Exam (Fall 2013)

Instructors: Adnan Waheed & Rabia Maqsood

December 12, 2013

Total Marks: 50	**Time Allowed: 3 hours**

Instructions:

(1) Understanding the question is part of exam. NO QUERIES WILL BE ENTERTAINED.
(2) **Think very carefully before attempting any of the questions.**
(3) Provide answers in the given space.
(4) Use back side of the paper if you need more space.
(5) Use answer sheet for rough work only. **No solutions will be checked on answer sheet.**
(6) Write neat & clean.
(7) Use permanent ink pens only.
(8) Poor programming approaches will decrease your marks.
(9) Think about the boundary conditions.

Roll No. _____Name:_____Section: _____

Question No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Total**
Marks	***09***	***04***	***04***	***05***	***09***	***10***	***04***	***05***	***50***

GOOD LUCK ☺

Question 1:	**Marks 06+03 = 09**

Consider a class **Stack**, which has following implemented member functions.

```
void push(int a)
int pop()
bool isEmpty()
bool isFull()
void reset()          // clears the stack.
```

a. Use this **Stack ADT** to implement **enqueue(int b)** and **int dequeue()** member functions of a **Queue ADT**. You may not use any other data structure objects (including arrays and lists) for implementation of these functions but may use additional temporary stack objects, if needed. ***(03 X 02 = 06)***

```
class QueueADT{
        Stack s1;
public:
        QueueADT ();
        void enqueue (int b);
        int dequeue ();
};
```

SOLUTION:

b. Analyze the running time of the queue operations. ***(1.5 X 02 = 03)***

Question 2:	**Marks 04**

Is the operation of deletion "commutative" in the sense that deleting *x* and then *y* from a binary search tree leaves the same tree as deleting *y* and then *x*? Argue why it is or give a counter example.

Question 3:	**Marks 04**

Place the given 4 words in an array of size 4 using the hash function given below. In case of collision, use linear probing.

DataSet = { float, int, char, break }


```
int HashFunction( char * word) {
    int length = strlen (word);              // returns the length of the word
    int val = 0;
    for (int i = 0; i< length; i+=2)
        val = val + AlphabeticalOrder (Word[i]) * 2;
  // AlphabeticalOrder returns the number of the letter in the English i.e. A=1, B=2, and so on.
    return val % maxsize;          }      //where maxsize is the number of words i.e. 4
```

Note: *Show your solution at the back side of this paper. No marks will be given for direct answer.*

Question 4:	**Marks 05**

Professor XYZ hypothesizes that substantial performance gains can be obtained if we modify the chaining scheme so that each list is kept in sorted order. How does the professor's modification affect the running time for successful searches, unsuccessful searches, insertions, and deletions?

Question 5:	**Marks 02+07 = 09**

Write the selection sort algorithm and provide code for sorting a doubly linked list using selection sort. Few assumptions are to be considered:

i. Values are already inserted in the list; do not write code for insertion.
ii. We have two pointers in doubly linked list: head and tail.
iii. Values at head and tail node are already sorted *(head has minimum value and tail has maximum value)*. You have to sort the internal nodes only.

Note: *You are required to change pointers for swapping operation. Do not swap the values of nodes.*

a. Algorithm:

b. Code:

Question 6:	**Marks 08+02 = 10**

You are assigned the task of planning the seating arrangement for Annual Prize Distribution Ceremony at *NUCES, FAST – Faisalabad Campus*, given a list of guests, V.

Suppose you are also given a lookup table T where $T[u]$ for $u \in V$ is a list of guests that u knows. If u knows v, then v knows u. You are required to arrange the seating such that any guest at a table knows every other guest sitting at the same table either directly or through some other guests sitting at the same table. For example, if x knows y, and y knows z, then x, y, z can sit at the same table.

a. Describe an efficient algorithm that, given V and T, returns the minimum number of tables needed to achieve this requirement.

NOTE: Specify all helper functions which you may use.

b. Analyze the running time of your algorithm.

Question 7:	**Marks 04**

Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Four students proposed the following search order (for different random values).

Student 1: **9, 85, 47, 68, 43, 57, 55**

Student 2: **10, 75, 64, 43, 60, 57, 55**

Student 3: **90, 12, 68, 34, 62, 45, 55**

Student 4: **79, 14, 72, 56, 16, 53, 55**

Discuss the correctness of all the proposed solutions.

Question 8:	**Marks 05**

Someone proposed to use circular linked list to implement a queue. Discuss this design choice and argue whether it is a good choice or not.