National University of Computer & Emerging Sciences, Karachi

Fall 2018 CS-Department

Mid Examination

22nd Oct 2018, 8:30 am – 10:30 am

Course Code: CL 201	**Course Name: Data Structures Lab**
Instructor Name / Names: Safia Baloch, Faizan Yousuf, Maham Mobin	
Student Roll No:	**Section: A**

Instructions:

- Return the question paper.
- Read question completely before answering it.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the sequence given in the question paper.

Time: 120 minutes. **Max Marks: 40**

Recursion: (20)

1. Write a recursive function of Collatz sequence. Which is related to a famous unsolved problem in number theory, known as Collatz problem or the 3n+1 problem.
   Note:
   The following table gives the sequences obtain for the first few values.

a_0	$a_0, a_1, a_2, \ldots$
1	1
2	2, 1
3	3, 10, 5, 16, 8, 4, 2, 1
4	4, 2, 1
5	5, 16, 8, 4, 2, 1
6	6, 3, 10, 5, 16, 8, 4, 2, 1

   consider the formula if next value obtain even or odd,

$$a_n = \begin{cases} \frac{1}{2} a_{n-1} & \text{for } a_{n-1} \text{ even} \\ 3\, a_{n-1} + 1 & \text{for } a_{n-1} \text{ odd} \end{cases}$$

2. Write a program Fibonacci, that takes an argument N and prints out the first N Fibonacci numbers using the following alternate definition. "DON'T USE POWER function"

```
F(n)    = 1                                  if n = 1 or n = 2
        = F((n+1)/2)^2 + F((n-1)/2)^2        if n is odd
        = F(n/2 + 1)^2 - F(n/2 - 1)^2        if n is even
```

   find the biggest Fibonacci number you can compute in under a minute using this definition? Comparing to Fibonacci(N-2)+Fibonacci(N-1). Hint "throw exception".

Sorting: (10)

3. Write a method called Bubble Sort that is a bi-directional version of bubble sort on an array of integers. That is, it makes alternating sweeps over the array, first going from the front to the end, then from the end to the front, and so on. When sweeping from front to end, it swaps forward an element that is larger than the element after it; when sweeping from end to front, it swaps backward an element that is smaller than the element before it. You may assume that the array is not null. The following diagram shows a run of this algorithm over a given array:

```
{16, 21, 45,  3, 11, 53,  8, 26, 49}

                  -->                sweep right
 16  21   3  11  45   8  26  49  53

                  <--                sweep left
  3  16  21   8  11  45  26  49  53

                  -->                sweep right
  3  16   8  11  21  26  45  49  53

                  <--                sweep left
  3   8  16  11  21  26  45  49  53

                  -->                sweep right
  3   8  11  16  21  26  45  49  53

                  <--                sweep left
```

Linked List: (10)

4. Create a single Link List and implement the following functions:
   a. Find Largest Data Node
   b. Delete List Largest Data Node