CS 2c03 Midterm Test (closed book)

February 26, 2020, 10:30-12:00 (90 minuts), Total: 104 pts

Instructions:

1. THIS EXAMINATION PAPER HAS **2** PAGES AND **11** QUESTIONS
2. WRITE YOUR NAME, STUDENT NUMBER IN THE RIGHT UPPER CORNER OF THE FIRST PAGE OF A BOOKLET.
3. Place all answers in the examination answer book provided.
4. The mark weighting for each question is given in the square brackets.
5. In your answers, be sure to convey (for each question): your understanding of the question being asked, your knowledge of what is expected, and evidence of your answer

1.[8] a.[4] Using only definition of O(f(n)) prove that the following statement is true:

$6n^3/(\log n + 1) = O(n^3)$

b.[4] Using only definition of O(f(n)) prove that the following statement is false:

$10n^3 + 9 = O(n)$

2.[15] Write a method ERASE_DUPLICANTS() to erase all the repetitions of all the elements of L. For instance if L = a,b,c,a,a,b,a the produced result should be a,b,c. Assume generic linked-list structure as described below.

What is the time complexity ? Justify your answer.

```
public class Queue<Item> implements Iterable<Item>
{
   private Node first; // link to least recently added node
   private Node last;  // link to most recently added node
   private int N;      // number of items on the queue

   private class Node
   {  // nested class to define nodes
      Item item;
      Node next;
   }

   public boolean isEmpty() {  return first == null;  }  // Or: N == 0.
   public int size()        {  return N;  }

   public void enqueue(Item item)
   {  // Add item to the end of the list.
      Node oldlast = last;
      last = new Node();
      last.item = item;
      last.next = null;
      if (isEmpty()) first = last;
      else           oldlast.next = last;
      N++;
   }

   public Item dequeue()
   {  // Remove item from the beginning of the list.
      Item item = first.item;
      first = first.next;
      if (isEmpty()) last = null;
      N--;
      return item;
   }
```

3.[7] Assume *Weighted Quick-Union with Path Compression* model and the following id[] array:

0	1	2	3	4	5	6	7	8	9
2	1	1	0	2	7	0	0	8	7

a.[1] Draw the forest represented by the above table.
b.[2] Execute union(1,8) and drew the resulting forest/tree
c.[2] Execute find(5) and drew the resulting forest/tree
d.[2] Execute find(7) and drew the resulting forest/tree

4.[8] A node of a binary tree is a *leaf* if it does not have any children, and it is *internal* if it is not a leaf. A binary tree is *full* if each internal node has exactly 2 children. Show that the number of leaves in a non-empty full binary tree is one more than the number of internal nodes.

5.[6] Show the *min-oriented heap* that results if the integers 6,4,7,3,1,9 are inserted into an empty heap. What is the result if *delete minimum*, i.e. delMin(), is then applied? *Show all steps*.

6.[16] Consider the sequence 6,4,7,3,1,9,2. Sort this sequence with (*show all steps*!)

a.[4] Insertion sort
b.[4] Top-down merge sort
c.[4] Quick sort with pivot set as the largest from the first two different values of the subarray.
d.[4] Quick sort with pivot set as the last element of the subarray.

7.[10] Prove that *Merge Sort* has worst case time complexity O(n log n). For simplicity assume $n = 2^k$ for some k.

8.[10] Consider the following problem. There is a sequence of records to be sorted in increasing order by their keys. The sequence is usually between 50000 to 100000 records and is almost always almost sorted, on average about 50 keys are out of places. These out of place keys have random values and random distribution.
Which sorting method would you recommend? Justify your answer.

9.[8] a.[4] Draw the Binary Search Tree (BST) that results when you insert the keys 6,4,7,3,1,9,2, in that order (associating the value i with the ith key, as per the convention in the text) into an initially empty tree. *Show all steps*! How many compares are needed to build the tree?
b.[4] Show the result of deletions, first 3 and next 4. Again *show all steps*.

10.[10] a[5] Draw the 2-3 tree that results when you insert the keys 6,4,7,3,1,9,2, in that order into an initially empty tree. *Show all steps*!
b.[5] Draw the red-black BST tree that results when you insert the keys 6,4,7,3,1,9,2, in that order into an initially empty tree. *Show all steps*!

11.[6] a[3] Demonstrate the insertion of the keys 11, 46, 52, 61, 34 into a hash table with collisions resolved by chaining. Let the table have 5 slots, and let the hash function be $h(k) = k \bmod 5$.
b[3] Repeat (a) using open addressing with auxiliary function $h'(k) = k \bmod 5$ and linear probing.