COMP 2140 Introduction to Data Structures

Cuneyt Akcora Final Exam (online) Winter 2020

LASTNAME, firstname: _____

Student number: _____

Directions:

- Type your answers in the spaces provided after the questions. Page 13 is extra space which you may use for any question. The backs of the pages are for rough work only.
- Classes for the coding questions are provided on the final two pages of the exam. These pages may be removed and used as reference. (Do NOT write answers on them.)
- The exam is out of 117 marks and contains 15 pages. You have three hours to write the exam.

Question	Mark	Out of
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
Total:		**100**

1- A sorting algorithm that sorts an array A is said to work in place if it orders the elements of A using only O(1) extra storage in addition to the O(n) space required to store A.

Name one sorting algorithm discussed in lecture that works in place.

Name one sorting algorithm discussed in lecture that does not work in place.

2- Suppose array A initially stores 10 ints in arbitrary order before executing the method partition (as defined in lecture for quick sort) such that the resulting ordering of array elements after a single pass of partition is as follows (after replacing the pivot to its proper position):

[1, 0, 3, 2, 4, 6, 5, 8, 7, 9]

Which element(s) could have been a pivot? Your answer may include more than one element.

3- Questions 3a to 3d refer to the following code.

```
public class ListNode<E> {
    private E data;
    private ListNode<E> next;

    public ListNode(E newData, ListNode<E> newNext) { data = newData;
        next = newNext;
    }

    ListNode<E> getNext() { return next; } E getData() { return data; }
}
public class ClassOne<E> { private ListNode<E> x;
    public ClassOne() { x = null; }
    public boolean test() { return (x == null); }
    public E methodA () {
        E result = null; if (!test()) {
            result = x.getData(); x = x.getNext();
        }
        return result;
    }
    public void methodB(E newData) { x = new ListNode<E>(newData, x); }
}
```

(a)[2] The class ClassOne implements the _____ADT (abstract data type).

(b)[2] Method test corresponds to the _____operation in this ADT.

(c)[2] Method methodA corresponds to the _____operation in this ADT.

(d)[2] Method methodB corresponds to the _____operation in this ADT.

4- Suppose the class Stack implements a stack of elements of a generic type E using the standard definitions for the methods push, pop, and isEmpty discussed in class. Questions 4a to 4c refer to the following code.

```
public class ClassTwo<E> { Stack<E> myStack;

    public ClassTwo() { myStack = new Stack<E>(); }
    public void methodC(E newData) { myStack.push(newData); }
    public E methodD() {
        Stack<E> tempStack = new Stack<E>(); E result = null;

        while (!myStack.isEmpty()) { result = myStack.pop(); tempStack.push(result);
        }

        if (!tempStack.isEmpty()) tempStack.pop();

        while (!tempStack.isEmpty()) myStack.push(tempStack.pop());

        return result;
    }
}
```

(a)[2] The class ClassTwo implements the _____ADT (abstract data type).

(b)[2] The method methodC implements the _____operation in this ADT.

(c)[2] The method methodD implements the _____operation in this ADT.

5- Short answers suffice. For Questions 5a, 5c, and 5e, you may not include the operations size or isEmpty in your answer.

a)[2] Name the three primary operations of the table (dictionary) abstract data type.

(b)[2] Name one data structure that is commonly used to implement a table (dictionary) efficiently.

(c)[2] Name the two primary operations of the queue abstract data type.

(d)[2] Name two data structures that are commonly used to implement a queue efficiently.

(e)[2] Name the two primary operations of the priority queue abstract data type.

(f)[2] Name one data structure that is commonly used to implement a priority queue efficiently.

6- Consider a table (dictionary) for Strings implemented as a hash table with collisions resolved using separate chaining and using the polynomial hash code.

(a)[5] Demonstrate how to compute the polynomial hash code of String “deaf”, using Horner’s method, with parameter a = 3.

Assume that the letter “a” casts to integer 0, “b” casts to 1, “c” to 2, “d” to 3, “e” to 4, “f” to 5, and so on.

Assume also that the table (dictionary) has size 7.

The powers of three are as follows: $3^0 = 1$, $3^1 = 3$, $3^2 = 9$, $3^3 = 27$, and $3^4 = 81$.

The final answer (i.e., the arithmetic) is unimportant — the sequence of operations you use to compute it is important.

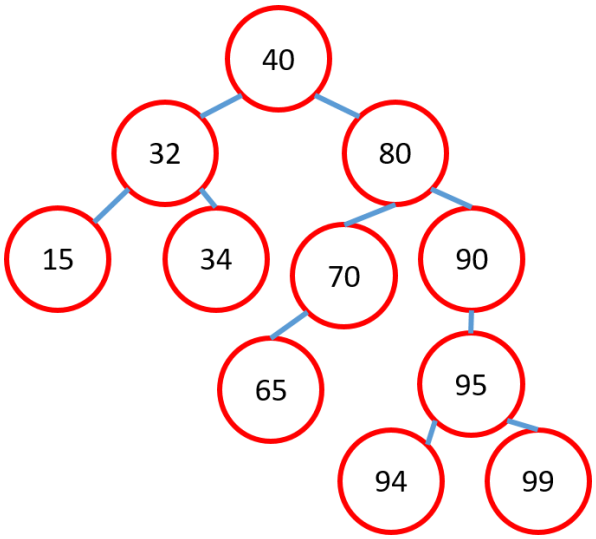
(b)[5] Assume the hash table has size 7, and that the hash function maps the following Strings as follows:

- “oak”, which hashes to 3;
- “pun”, which hashes to 5;
- “dab”, which hashes to 0;
- “hop”, which hashes to 1;
- “wit”, which hashes to 5; and
- “yes”, which hashes to 1.

Draw the hash table that is the result of inserting the above Strings in the order given into an initially empty hash table, with collisions resolved using chaining.

7- Draw the binary search tree (by using Word shapes) that results after inserting the following sequence of keys (in the order given), assuming the tree was initially empty: [6, 4, 1, 7, 3, 5, 2].

8- Each of the following questions refers to the following binary tree:



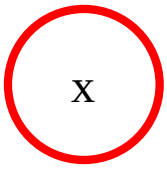
(a) [4] List the sequence of nodes visited during a pre-order traversal of this tree

(b) [4] List the sequence of nodes visited during a post-order traversal of this tree.

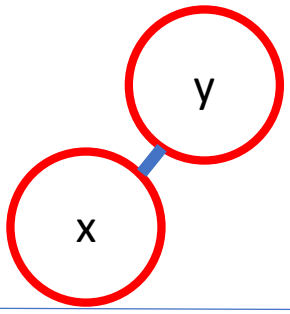
(c) [2] What node in the tree has the most ancestors?

9- Perform leaf-based 2-3 tree insertions to insert 3, 10, 2, 8, 5, 6, 7, 1, 4, 9 (in that order) into an empty leaf-based 2-3 tree, drawing the tree **after each insertion** is completed. (This question will be answered by drawing shapes (nodes and edges) in Office Word). For your convenience, you can use the following shapes to draw.

Insertion 1



Insertion 2



Additional page left for question 9.

10- Does the following array store a binary tree in maximum heap ordering? Why or why not?

9	8	4	5	6	2	1

11- Suppose a graph G1 has n vertices, and each vertex is adjacent to 4 other vertices in G1. Suppose a second graph G2 has n vertices, and each vertex is adjacent to n − 2 other vertices in G2. Suppose furthermore that an algorithm needs to check whether there is an edge from a vertex v1 to a vertex v2 in both G1 and G2.

Select the true statement by marking it with a cross (as [x]). If the graphs G1 and G2 are stored using adjacency lists, then in the worst case

[ ]it requires longer to check whether the edge exists in G1 than whether it exists in G2

[ ] it requires longer to check whether the edge exists in G2 than whether it exists in G1

[ ] it requires similar time to check whether the edge exists in G1 and G2

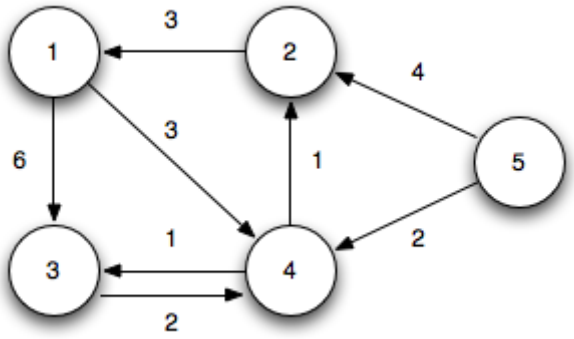
[ ] none of the above

12- Fill in the body of the method checkHeap such that it returns true if and only if the integers stored in array h are in maximum heap ordering according to the usual array representation for storing the keys of a binary heap. You may assume that array h has been properly initialized, that h.length returns its size, and that the heap is full (i.e., all array values correspond to elements stored in the heap).

```
public static boolean checkHeap( int [] h ) {
}
```

13- Given the following graph, find the shortest path between vertices 1 and 2 by using the Dijkstra's algorithm. At each time a new vertex is finalized, write its id, distance and previous vertex. For example,
Step 1: added vertex id=1, distance=0, previous vertex=1
Step 2: added vertex id=1, distance=0, previous vertex=1

And so on.



14- The following undirected weighted graph is given with and edge list. An edge 1-3:4 implies that vertices 1 and 3 are connected by an edge weight of 4. Use the Prim's algorithm to find the spanning tree starting from node 1.

A- Give the order of vertices added by the Prim's algorithm. For example, [1,5,6,7….].

B- Give the total weight of the edges on the shortest path:

1-3:4,
2-3:5,
3-5:2,
5-7:1,
6-8:3,
3-9:2,
8-9:4