King Fahd University of Petroleum & Minerals

College of Computer Science and Engineering

Information and Computer Science Department

First Semester 121 (2012/2013)

ICS 202 – Data Structures

Major Exam 1

Saturday, 6th October, 2012

Time: 90 minutes

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID# |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section 02 |  | Question # | Max Marks | Marks Obtained |
| Dr. Wasfi |  | 1 | 20 |  |
| 10-10:50am |  | 2 | 25 |  |
| Section 05 |  | 3 | 25 |  |
| Dr. Sami |  | 4 | 30 |  |
| 9-9:50am |  | Total | 100 |  |

**Instructions**

1. **Write your name and ID in the respective boxes above and circle your section.**
2. **This exam consists of 8 pages, including this page, plus one double-sided reference sheet, containing 4 questions.**
3. **You have to answer all 4 questions.**
4. **The exam is closed book and closed notes. No calculators or any helping aids are allowed.**
5. **Make sure you turn off your mobile phone and keep it in your pocket if you have one.**
6. **The questions are not equally weighed.**
7. **The maximum number of points for this exam is 100.**
8. **You have exactly 90 minutes to finish the exam.**
9. **Make sure your answers are readable.**
10. **If there is no space on the front of the page, feel free to use the back of the page. Make sure you indicate this in order not to miss grading it.**

Q.2: (30 points) Consider the SLL class definition as seen in the lectures:

**public class SLL<T> {**

 **protected SLLNode<T> head, tail;**

 **public SLL() {**

 **head = tail = null;**

 **}**

 **public boolean isEmpty() {**

 **return head == null;**

 **}**

 **...**

**}**

1. [15 pts] Implement a method called **kSmallest** that replaces the first k smallest values in the beginning of the linked list.

 For example, if a linked list of Integer elements is containing the following values,

25

7

14

-1

6

34

2

**null**

**head**

**tail**

After a call to:

**KSmallest(3)**, the single list becomes:

-1

2

6

25

7

14

34

**null**

**head**

**tail**

**Public kSmallest(int k){**

**}**

1. [10 pts] Compute the big-O complexity of this method and express it in terms of n (size of the linked list) and the parameter k.

**Important:** Show all steps of computation (summations, etc.).

1. [5 pts] If the best, average and worst cases depend on the value of k, provide the complexity of the **kSmallest** method in each one of the cases.

Q.3: [20 pts]

1. Write a recursive method printVertical that takes as input a positive integer and prints its digits in vertical.

For example the output of the call: **printVertical(2849)** is:

9

4

8

2

**public void printVertical(int n){**

}

1. Is the implemented method tail-recursive or non-tail recursive? Justify your answer.
2. Show the trace of the call **printVertical**(51427895)

**Quick Reference Sheet**

|  |  |
| --- | --- |
| public interface Iterator { boolean hasNext( ); Object next( ) throws NoSuchElementException;}public interface Visitor { void visit (Object object); boolean isDone( );}public interface Container { int getCount( ); boolean isEmpty( ); boolean isFull( ); void purge( ); void accept (Visitor visitor); Iterator iterator( );}public interface SearchableContainer extends Container { boolean isMember (Comparable object); void insert (Comparable object); void withdraw (Comparable obj); Comparable find (Comparable object);}public class Association implements Comparable public Association(Comparable key, Object val) public Association(Comparable key) public Comparable getKey( ) public Object getValue( ) public void setKey(Comparable key) public void setValue(Object value ) public int compareTo(Object obj) public boolean equals(Object obj) public String toString( )} | public class MyLinkedList { public void purge( ) public Element getHead( )  public Element getTail( )  public Element find(Object obj) public boolean isEmpty( )  public Object getFirst( ) public Object getLast( ) public void prepend(Object obj) public void append(Object obj)  public void assign(MyLinkedList list)  public void extract(Object obj) public void extractFirst( ) public void extractLast( ) public String toString( ) public Iterator iterator( )  public final class Element { public Object getData( ) public Element getNext( ) public void insertAfter(Object obj)  public void insertBefore(Object obj) public void extract( )  }}public interface Stack extends Container { Object getTop( ); void push(Object obj); Object pop( );}public interface Queue extends Container { Object getHead( ); void enqueue(Object obj); Object dequeue( );} |

 ,  , , 