University of Glasgow

XXXday May XX, 2011
XX.XX am/pm – XX.XX am/pm
(Duration: X hour XX minutes)

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Algorithms and Data Structures 2

(Answer all questions.)

This examination paper is worth a total of 50 marks

You must not leave the examination room within the first hour or the last half-hour of the examination. (for exams of 2 hours duration)

or

You must not leave the examination room within the first half hour or the last fifteen minutes of the examination. (for exams of less than 2 hours duration)

1. A binary search tree is a binary tree T such that each node of T stores an item e. Items stored in the left subtree rooted at a node v are less than the item in node v, and items stored in the right subtree rooted at a node v are greater than the item in node v. Below is java code for the BNode class, and below that code for the BSTree class.

```
public class BNode {
    private int   item;
    private BNode left;
    private BNode right;

    public BNode(int e, BNode left,BNode right){
       this.item  = e;
       this.left  = left;
       this.right = right;
    }

    public int   getItem(){return item;}
    public BNode getLeft(){return left;}
    public BNode getRight(){return right;}
    public void  setLeft(BNode nd){left = nd;}
    public void  setRight(BNode nd){right = nd;}
}


public class BSTree {
    private BNode root;

    public BSTree(){root = null;}

    public BNode root(){return root;}
    public boolean isEmpty(){return root == null;}

    public void insert(int e){
       if (isEmpty()) {root = new BNode(e,null,null);}
       else insert(e,root);
    }

    private static void insert(int e,BNode nd){...}

    public boolean isPresent(int e){return root != null && isPresent(e,root);}

    private static boolean isPresent(int e,BNode nd){...}

}
```

(a) Write java code for the method **insert(int e,BNode nd)** in class BSTree, where the method inserts the integer **e** into the tree if and only if **e** is not already present in the tree. **[5]**

(b) Write java code for the method **isPresent(int e,Bnode nd)**, where the method delivers true if and only if **e** is in the tree. **[5]**

(c) Assume that the following items are inserted into an empty BSTree in the following order: 30, 40, 24, 58, 48, 26, 11, 13, 36.

- Draw the tree.
- What is the height of the tree?
- Write out the preorder, inorder and postorder traversals of the tree. **[5]**

(d) Draw the tree after the node with item 30 has been deleted and outline the algorithm you used for the deletion (you do not need to write Java code). **[3]**

2. A stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle. In the class definition below for Stack, the stack is implemented as an array of Objects.

```
public class Stack {

    private Object[] S;
    int tos = -1; // top of stack pointer
    int capacity;

    public Stack(int capacity){
      this.capacity = capacity;
      S = new Object[capacity];
    }

    public int size(){...}
    public boolean isEmpty(){...}
    public boolean isFull(){...}

    public void push(Object o) throws StackException {...}
    public Object top() throws StackException {...}
    public Object pop() throws StackException {...}
}


public class StackException extends RuntimeException {
  public StackException(){}
  public StackException(String msg){super(msg);}
}
```

(a) Give a Java implementation for the undefined methods above (**size, isEmpty, isFull, push, top, pop**) taking care to address stack overflow and stack underflow. **[8]**

(b) The stack could have been implemented using a singly linked list (rather than a one dimensional array as above). What would be the relative advantages and disadvantages of each of those implementations? **[4]**

3. The dictionary abstract data type stores key-element pairs (k,v), which we call entries, where k is the key and v is the value. A dictionary allows for multiple entries with the same key, much like an English dictionary, where we can have multiple definitions of the same word. The primary use of a dictionary is to store values so that they can be located quickly using keys. We might realize a dictionary using an unordered linked list or a sorted array. Given a dictionary D, what would be the complexity of the following operations when D is realized using an unsorted linked list and when D is realized using a sorted array?

```
D.insert(k,v)     D.find(k)
D.size()          D.remove(k)
```

Explain each one of your answers, and state any assumptions that you make**.**
NOTE: D.find(k) returns an arbitrary entry (k,v) whose key is equal to k, or null if no such entry exists **[8]**

4. A map abstract data structure allows us to store key-value pairs (k,v) , but is different from the dictionary abstract data type (above) because each key is unique. Consequently the association of keys to values defines a mapping. We might implement the map abstract data type using a hash table.

   (a) What is a hash table and why would we want to use one? **[2]**

   (b) What is a hash function and what properties would we like it to have? **[2]**

   (c) What is a "collision"? **[1]**

   (d) Describe the "separate chaining" method for hash collision resolution. **[2]**

   (e) Describe the "linear probing" method for hash collision resolution, taking into consideration the methods insert, find, and remove **[5]**