COURSEWORK ASSIGNMENT **UNIVERSITY OF EAST ANGLIA**

School of Computing Sciences

MODULE : Data Structures and Algorithms

ASSIGNMENT TITLE : Coursework 1

DATE SET **:** 02/11/16

DATE OF SUBMISSION **:** 08/12/16

RETURN DATE **:** 10/01/17

ASSIGNMENT VALUE **:** 15%

SET BY **:** Geoff McKeown **SIGNED:**

CHECKED BY **:** Anthony Bagnall **SIGNED:**

Aim:

Subject specific

To test your ability to design, analyse and implement algorithms.

Transferable skills

Describing algorithms using pseudo-code; use of mathematical notation when analysing algorithms; computer programming.

Learning outcomes:

Subject specific

Increased experience of programming in Java; increased awareness of the importance of algorithm complexity.

Assessment criteria:

Part marks are shown on the exercise sheet.

Description of the Assignment

1 Design and Analysis (30%)

This question requires you to design, analyse and implement an algorithm to perform the following task.

Given an array of size n, where each element of the array is an integer in the range 1 to 100 inclusive, create and display a histogram that allows you to inspect visually the frequency distribution of the collection of integers, producing a chart similar to the one below:

1-10	***
11-20	**********
21-30	********
31-40	******
41-50	********
51-60	*******
61-70	**********
71-80	****
81-90	
91-100	*

1. Design and write an algorithm in pseudo-code to solve this problem. **[15 marks]**
2. Determine the run-time complexity of your algorithm. State the order of your run-time complexity function. **[5 marks]**
3. Implement and test your algorithm. **[10 marks]**

2 Design and Implementation of a Recursive Algorithm(50%)

Sometimes, we wish to examine all permutations of a collection of n distinct elements. For example, the permutations of the elements a, b, c are

$$abc, acb, bac, bca, cab \text{ and } cba.$$

The number of permutations of n elements is $n!$

Let $A = \langle a, b, c \rangle$ and let A_i denote the collection obtained from A by deleting the i^{th} element. Thus,

$$A_1 = \langle b, c \rangle, \qquad A_2 = \langle a, c \rangle, \qquad A_3 = \langle a, b \rangle.$$

For a collection of distinct elements, X, define perm(X) to be the set of permutations of the elements in X. Thus,

$$\text{perm}(A_1) = \{bc, cb\}, \qquad \text{perm}(A_2) = \{ac, ca\}, \qquad \text{perm}(A_3) = \{ab, ba\}.$$

For $X = \langle x_1, x_2, \ldots, x_n \rangle$ we define $e.X = \langle ex_1, ex_2, \ldots, ex_n \rangle$. Then

$$a.\text{perm}(A_1) = \{abc, acb\}, \qquad b.\text{perm}(A_2) = \{bac, bca\}, \qquad c.\text{perm}(A_3) = \{cab, cba\}.$$

Note that $\text{perm}(A) = a.\text{perm}(A_1) \cup b.\text{perm}(A_2) \cup c.\text{perm}(A_3)$.

1. Design a recursive algorithm based on the above ideas to determine a list of (string representations of) all of the permutations of a given list of n distinct elements. **[25 marks]**

   [Assume that there is a method, `toString`, for converting an element into its string representation.]

2. Implement your algorithm in Java. **[25 marks]**

3 Exercise on Balanced Binary Search Trees (20%)

Draw diagrams to illustrate your answers to this exercise.

Insert the following integers in the order given into a BST. Do not perform any re-balancing operations.

$$37, 25, 45, 22, 39, 19, 30, 50, 27, 34$$

Insert 28 into the resulting tree and illustrate the operations necessary to re-balance the tree.

Now delete 34 from the resulting tree and again illustrate the operations necessary to re-balance the tree.

Hand drawn trees are acceptable.

Assessment

Marks will be awarded as follows

30% Question 1

50% Question 2

20% Question 3

 Submission instructions overleaf.

Submission Procedure

To the hub: A hard copy of your solutions to all questions, including code listings and listings of your test results.

via blackboard: a zipped file containing Netbeans projects with your code for the practical questions.

Plagiarism: Plagiarism is the copying of close paraphrasing of published or unpublished work, including the work of another student; without due acknowledgement. Plagiarism is regarded a serious offence by the University, and all cases will be investigated. Possible consequences of plagiarism include deduction of marks and disciplinary action, as detailed by UEA's Policy on Plagiarism and Collusion.