CE204-5-SP

UNIVERSITY OF ESSEX

Undergraduate Examinations 2015

DATA STRUCTURES AND ALGORITHMS

Time allowed: **TWO** hours

Candidates must answer **ALL** questions.

The paper consists of **FOUR** questions.

The questions are of equal weight.

The percentages shown in brackets provide an indication of the proportion of the total marks for the **PAPER** which will be allocated.

*Candidates must answer **ALL** questions.*

Question 1

(a) (i) Prove directly from the definition of big O that $5n^2+9n+7$ is $O(n^2)$. [3%]

(ii) State with reasons a big O estimate for the worst-case time complexity of the following function. [5%]

```
int myfunc(int n)
{ int result = 0;
  for (int i = 0; i<n; i++)
    for (int j = i; j>0; j--)
      if (i%j == 0)
        result += j;
  return result;
}
```

(b) Provide a complete Java class that implements the interface [17%]

```
interface StringStack
{ boolean isEmpty();
  void push(String s);
  String top();
  void pop();
}
```

The class should provide an implementation of a first-in-last-out stack and use a linked list to store the contents of the stack. The linked list must be implemented using list cells (i.e. you are not allowed to use the `LinkedList` class from the collections framework); the cell class should be written as an inner class. If applied to an empty stack the `pop` and `top` methods should throw an exception – you may assume that an appropriate `StackException` class has been declared. A no-argument constructor should be provided to ensure that a newly-created stack will be empty.

.

Question 2

(a) Explain what is meant by a *non-computable* function, and give an outline of a proof that such a function exists. [10%]

(b) Describe the *insertion sort* algorithm and comment briefly on its time complexity. [7%]

(c) The following interface specifies the binary tree type. [8%]

```
interface BinaryTree<T>
{ boolean isEmpty();
  T rootValue();
  BinaryTree<T> leftChild();
  BinaryTree<T> rightChild();
}
```

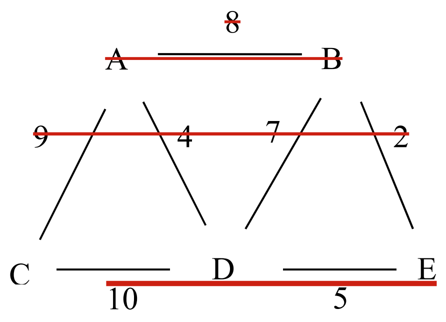
Write a method that takes an argument of type `BinaryTree<Float>` and uses a pre-order traversal to calculate the sum of all of the numbers in the tree specified in the argument and return this sum as a value of type `float`.

~~Question 3~~

~~(a) Explain what is meant by the term *connected* as used to describe undirected graphs.~~ ~~[2%]~~

~~(b) Describe *Kruskal's algorithm* for finding shortest paths in a connected undirected graph.~~ ~~[10%]~~

~~(c) Show, step by step, the use of Kruskal's algorithm to find a minimum cost spanning tree for the graph shown below. At each stage you should show the connection sets.~~ ~~[13%]~~



~~**Question 4**~~

~~(a) Explain the meaning of the terms *balanced* and *AVL-balanced* as used to describe binary trees.~~ ~~[4%]~~

~~(b) Give examples of binary search trees containing exactly 5 nodes that are (i) neither balanced nor AVL-balanced, (ii) AVL-balanced but not balanced, and (iii) both balanced and AVL-balanced.~~ ~~[3%]~~

~~(c) Show, step by step, the results of inserting the following numbers (in the order in which they are listed) into an initially-empty binary search tree, using the AVL rebalancing algorithm when necessary in order to ensure that the tree is AVL-balanced after each insertion.~~ ~~[18%]~~

2 23 8 33 45 11 17

END OF PAPER CE204-5-SP