Algorithms & Data Structures (M)

Assessed Exercise (2019–2020)

This exercise is assessed. Its weighting is *0.3* in the ADS(M) course assessment. There are 2 parts of the assessment. Read **all three pages** of this document thoroughly.

The deadline is *Monday 16 March 2020* at 16:30.

1. [5 marks]

This part of the exercise is to show that you understand the Java interface `Set` and it's implementation, `TreeSet`. See Lecture 1 for an example of these classes in use.

You have been provided with three files: a text file `birds.txt`, an outline of a class, `AssEx1_outline.java`, and a pdf file `exampleOutput.pdf`. You should make a copy of `AssEx1_outline.java` and rename it `AssEx1.java`.

Complete the `AssEx1.java` class as instructed in the comments within the file. Do **not** change the `main` method in any way or include any package declaration. You should include any import statements necessary for me to run the program from the command line thus:

```
> javac AssEx1.java
> java AssEx1 "birds.txt"
```

You may add helper methods if you need to, but any unnecessary complexity (of code) will be penalised.

When you are happy with your class, run it and store the output in a pdf document called `AssEx1_output.pdf` (remove any additional print statements you may have added to your methods for testing purposes first). You should inlcude your name and matriculation number at the top of this document.

The file `exampleOutput.pdf` which I generated using a different (smaller) input text file will illustrate the exact form of the output required. You can use whichever font you like.

Full submission instructions will be included at the end of this document, but for this part of the exercise you will submit your two files: `AssEx1.java` and `AssEx1_output.pdf`.

2. **[25 marks]**

This part of the exercise is to show that you understand:

- the difference between an algorithm and its Java implementation, and
- how to manipulate Singly Linked Lists

2.1 [17 marks] Devise algorithms to do the following:

- Insert a new node into the **end** (i.e. not the front, unless the current list is empty) of a linked list L with first node `first`
- Delete every alternate node from a linked list L with first node `first.` For example, if L consists of nodes containing the following strings:

  `ant, bat, cat, dog, elephant`

  your algorithm should change the list so that it now consists only of the nodes containing the strings:

  `ant, cat, elephant`
- Merge three sorted linked lists L, L1 and L2 preserving the order and removing duplicates to create a new linked list L3. You may refer to a helper algorithm, but must include that algorithm in your solution.

Use a suitable word processing package to create a document: `AssEx1Algorithms.pdf` containing, for each algorithm:

- a picture illustrating your algorithm (this may be a photograph/scan of a pen and paper drawing)
- a formal step-by-step algorithm
- an analysis of the complexity of your algorithm, in terms of the lengths of any linked lists involved.

2.2 [8 marks] Implement your algorithms as Java methods by following the instructions below. You have been provided with an outline Java class for this part of the exercise: `AssessmentSLL_outline.java`.

You should:

- make a copy of this file and rename it: `AssessmentSLL.java`.
- complete the `AssessmentSLL.java` class as instructed in the comments within the file. Do **not**
  - change the provided methods in any way or
  - include any package declaration.
  - Do not change the names of any methods in the class.

  But you may include additional helper methods if necessary.
- Create your own test program, `AssExTest.java`, say, which uses your class (but you will not submit this class). I should be able to run my own test program, which will use your `AssessmentSLL.java` class, from the command line thus:

```
> javac AssessmentSLL.java
> javac AssExTest.java
> java AssExTest
```

For this part of the exercise you will submit your two files: `AssessmentSLL.java` and `AssEx1Algorithms.pdf.`

Submission

By the deadline stated above, you must submit your deliverables through the ADS(M) Moodle page. (Click “Assessed exercise 2 submission”.)

The deliverables are all the files requested in parts 1 and 2, namely:

Part 1: `AssEx1.java` and `AssEx1_output.pdf`
Part 2: `AssessmentSLL.java` and
`AssEx1Algorithms.pdf`

So not submit any additional files.

Your files should be contained in a single (zipped) folder. Penalties will be applied for failure to comply with submission instructions (e.g. those that include additional files, package statements etc.) These include instructions contained within the comments of the outline Java files.