

CIS2344 Algorithms Processes & Data

Assignment 2018/2019

Logbook

Hugh Osborne

September 11, 2018

Written by:	Hugh Osborne
Last update:	September 11, 2018
Hand out:	17.15, Tuesday 25/9/2018
Hand in Weeks 1–5:	17.15, Tuesday 13/11/2018
Hand in Weeks 7–11:	17.15, Tuesday 8/1/2019
Hand in Weeks 13–16:	17.15, Tuesday 26/2/2019
Hand in Weeks 17–22:	17.15, Tuesday 9/4/2019
Hand back:	17.15, Tuesday 7/5/2019
Hand in tutor referral:	17.15, Tuesday 21/5/2019

Contents

1 Assignment specification	1
1.1 Aims	1
1.2 Outcomes to be assessed	2
1.3 Logbook requirements	2
1.4 Self assessment	3
1.5 Group component	3
1.6 Resources required	3
1.7 Hand in	3
1.8 Assessment criteria	3
2 Feedback forms	7
2.1 Mark allocation	7
2.2 Weekly exercises	7
2.3 Logbook	11
2.4 Self-assessment	12
2.5 Additional work	13

1 Assignment specification

1.1 Aims

- To develop familiarity with standard algorithms and data structures — linear data structures, trees and graph structures, sorting and searching.
- To develop an understanding of basic concepts and features of concurrency — semaphores, shared memory, message passing, monitors, threads; deadlock, starvation, non-determinism.

- To develop an awareness of principles of quantum computing.
- To develop an understanding of approaches to proving code correct.
- To develop an understanding of the issues that affect code efficiency.

1.2 Outcomes to be assessed

1. Discuss the classification of algorithms according to efficiency and complexity.
2. Prove code correct.
3. Describe a range of useful algorithms and the properties and uses of common data structures.
4. Demonstrate a knowledge of the characteristics of a range of concurrency paradigms.
5. Explain the differences between classical and quantum computing.
6. Develop complex algorithms and data structures to solve practical problems.
7. Use a standard notation to analyse the efficiency and complexity of algorithms.
8. Implement working concurrent software using appropriate programming language semantics.

1.3 Logbook requirements

You are required to develop a logbook of work undertaken in this module. Your logbook will be developed throughout the year. There are no major restrictions on how you should develop your logbook except that it should be a tool that will assist your learning on this module. Basically, it is a record of your work in the module. In each set of exercises one or more questions will be identified as a requirement for inclusion in your logbook, but you are free to include other work if you feel that this will improve the logbook. Also, most weeks will have one or more exercises that will be identified as “model” exercises. These exercises have been selected as being similar to the logbook exercises. Model answers will be published for the model exercises one week after the last scheduled practical for that set of exercises. For example, the first set of exercises covers weeks 1 & 2. The last scheduled practical for this set of exercises will therefore be the practical at 16.15 on Tuesday of week 3¹. The model answer for the week 1 & 2 model question will therefore be published at 17.15 on Tuesday of week 4. i.e. Tuesday 16th October. Since model answers are being provided for the model questions there is no point in including these in your logbook (unless you believe your answers are better than the model answers, in which case please *do* include them, explaining *why* they are better).

The logbook will be assessed on its completeness and use as a source of reference. You will be expected to hand in your logbook in the following stages:

- Weeks 1–5: 17.15, Tuesday 13/11/2018
- Weeks 7–11: 17.15, Tuesday 8/1/2019
- Weeks 13–16: 17.15, Tuesday 26/2/2019
- Weeks 17–22: 17.15, Tuesday 9/4/2019

¹The lecture for this module is timetabled *after* the practicals. This means that the practicals for weeks 1 & 2 will take place in weeks 2 & 3, etc.

1.4 Self assessment

You will also be required to assess your own work. For each set of logbook exercises you should assess your performance using grades **1** to **5**, according to the grading scheme available on Brightspace, a version of which can also be found in section 2.2.

Please note that for a good mark for the self-assessment it is not sufficient to simply complete the forms. You must also justify your assessment — i.e. provide evidence to back up your assessment.

1.5 Group component

There is no group component — this is an individual piece of work. All work must be your own. Collusion is cheating and will be dealt with severely. Any doubts as to the originality of any work submitted will be reported to the Academic Integrity Officer who will hold a full investigation.

1.6 Resources required

Access to the module pages on Brightspace. Access to a computer with Java. Access to a computer with IntelliJ or similar IDE is advisable.

1.7 Hand in

The deadlines for submitting your logbook will be as detailed above. Submission will be through a link under the “Assignments” tab on Brightspace. You should submit your logbook as a single .pdf document. You do *not* need to submit your source code files.

1.8 Assessment criteria

The logbook is the only assessment for this module, and therefore forms 100% of the marks for this module.

The logbook will be marked out of 100%. This mark will be composed of:

- 60% for the work on the identified logbook exercises;
- 20% for the quality of your self-assessment of your work and engagement with the module
- 20% for the overall quality of your logbook

In addition, you can earn up to 10% compensation for any shortcomings in the above core aims of the assignment through work on exercises beyond the identified logbook exercises.

The following grade descriptors give an idea of the standard of work required to achieve each grade:

1.8.1 Grade A+ 90-100%

Logbook will satisfy all, or almost all, of the criteria for a Grade A (section 1.8.2) and will, furthermore, demonstrate excellence throughout — e.g.:

- Will almost certainly contain a comprehensive set of the additional questions from the practicals
- Will probably contain a significant amount of work from outside the exercises provided, e.g. from textbooks, or from other sources such as Project Euler.
- All, or almost all of the logbook could be used as an exemplar source of reference

1.8.2 Grade A 80-89%

Logbook will satisfy all, or almost all, of the criteria for a Grade A— (section 1.8.3) and will, furthermore, demonstrate excellence in many sections — e.g.:

- Is very likely to contain a comprehensive set of the additional questions from the practicals
- May contain some additional work outside the exercises provided, e.g. from textbooks, or from other sources such as Project Euler.
- Clear that code has been thoroughly tested, using appropriate techniques. Theoretical analyses are strong.
- Many parts of the logbook could be used as an exemplar source of reference

1.8.3 Grade A— 70-79%

Logbook will satisfy all, or almost all, of the criteria for a Grade B (section 1.8.4) but will demonstrate excellence in some sections — e.g.:

- Is likely to contain answers to some of the additional questions from the practicals
- Will contain convincing evidence of well thought out and extensive testing strategies for coding exercises, and of thorough analysis where required
- Parts of the logbook could be used as an exemplar source of reference

1.8.4 Grade B 60-69%

Logbook will satisfy all, or almost all of the criteria for a Grade C (section 1.8.5) but will demonstrate above average work in some sections — e.g.:

- Will contain a good set of answers to the flagged logbook exercises
- Will contain stronger evidence of testing and analysis than at grade C
- May contain some of the additional exercises from the practicals
- Logbook will be focused on the value of the materials as a source of future reference

1.8.5 Grade C 50-59%

This would be the average grade for a typical logbook with certain strengths but also some weaknesses — e.g.:

- Logbook will be well organised, use separators or links to show organisation, have an index and/or list of contents
- Will contain student's own notes to aid understanding
- Will contain answers to most of the flagged logbook exercises
- Will contain some evidence of testing and analysis as appropriate
- Logbook will generally be usable as a source of reference

1.8.6 Grade D 40-49%

Logbook will satisfy most of the criteria for a Grade C (section 1.8.5) but may be deficient in some way, such as:

- Logbook may not be well organised
- Logbook may not be useful as a source of future reference
- May be some gaps in testing and analysis
- Logbook may not be up-to-date
- Some answers to tutorial/practical exercises will be missing
- There will be evidence of work in progress and some working programs

1.8.7 Grade E 30-39%

The logbook is inadequate in some respects, but should be relatively easily recoverable:

- Some effort shown
- Some evidence of understanding
- Some evidence of basic testing and analysis

This grade is eligible for a tutor referral.

1.8.8 Grade F+ 21-29%

The logbook is inadequate in quite a few respects, but may be recoverable, given sufficient effort:

- Poor logbook
- Not much evidence of understanding
- Not much evidence of testing and analysis
- Not much evidence of learning
- Not really adequate at second year level

This grade is eligible for a tutor referral.

1.8.9 Grade F 11-19%

The logbook is inadequate in many respects, but could be recoverable, given major effort:

- Very poor logbook
- Very little evidence of understanding
- Very little evidence of testing and analysis
- Very little evidence of learning
- Clearly not adequate at second year level

This grade is eligible for a tutor referral.

1.8.10 Grade F – 1-9%

The logbook is inadequate in all, or almost all respects, and is unlikely, but not impossible, to be recoverable

- Extremely poor logbook
- Practically no evidence of understanding
- Practically no evidence of testing and analysis
- Practically no evidence of learning
- Would be inadequate, or barely adequate at *first* year level

This grade is eligible for a tutor referral.

1.8.11 Grade G 0%

Logbook missing. Tutor referral is *not* available for students who fail to submit any work by the deadline (once any approved extensions and/or recognised extenuating circumstances are taken into account).

2 Feedback forms

2.1 Mark allocation

There are sixteen identified (sets of) logbook exercises, submitted in four stages. Each (set of) exercise(s) will each be marked out of five. The worst mark at each stage will be discounted², leaving 60% to be earned from the exercises themselves. The self-assessment exercises will be worth 20%, and the remaining 20% will come from the overall logbook quality³. You also have the opportunity to compensate for shortcomings in the overall mark by submission of additional work⁴.

2.2 Weekly exercises

The following tables show you the criteria that will be used to mark each set of exercises.

2.2.1 Submission 1: 17.15, Tuesday 13/11/2018

Submission 1: 17.15, Tuesday 13/11/2018					
5	4	3	2	1	0
Weeks 1&2: Searcher					
Near Javadoc compliant. Full set of automated tests		Good documentation	Working searcher	Some attempt	NS
Weeks 3&4: Generic swap					
Full automated testing	Tested	Correct generic method	Some use of generics	(Non-generic) swap implemented	NS
Week 5: Sorting					
Big O analysis, or near equivalent	Extensive testing. Good analysis	Additional algorithms. Larger data set	Graphic presentation. Some comparison	Basic timing results for methods provided. No analysis.	NS

²E.g.: the first submission stage is in week 8, when three sets of exercises will be submitted, from weeks 1 & 2, from weeks 3 & 4, and from week 5. These will be marked, and the lowest of the three marks will be discounted.

³Please note that in the feedback tables below code documentation and testing appear both as marking criteria for some of the weekly exercises and as partial determiners of overall logbook quality. This is deliberate.

⁴This means that the assignment is effectively marked out of 110%, with a 100% cap. This is also deliberate.

2.2.2 Submission 2: 17.15, Tuesday 8/1/2019

Submission 2: 17.15, Tuesday 8/1/2019					
5	4	3	2	1	0
Week 6: Linked lists					
Boundary checking and exceptions. Test suite		All methods implemented	add or delete implemented	Some attempt	NS
Week 8: Binary trees					
Good documentation. Good testing		All methods implemented	insert or inOrderTraversal implemented	Some attempt	NS
Week 9: Hashtables					
Full analysis	Discussion of memory management issues	Analysis of how internal array slots are allocated	Some description of inner workings of hashtables	Little more than a sequence of screenshots	NS
Week 10: Tree traversal					
Javadoc documentation. Full testing (e.g. sorted in-order traversal)		Full implementation	Partial implementation	Some attempt	NS
Week 11: Topological sort					
Javadoc documentation. Full testing (e.g. nodes/successors in correct order)		Full implementation	Partial implementation	Some attempt	NS

2.2.3 Submission 3: 17.15, Tuesday 26/2/2019

Submission 3: 17.15, Tuesday 26/2/2019					
5	4	3	2	1	0
Week 13: Counters					
Good analysis of termination question	Clear explanation of min & max values	Evidence of multiple runs. Correctly identifies min & max values	Evidence of implementation. Some answers to questions, but lacking analysis	Little evidence of implementation. Little attempt to answer questions	NS
Week 14: Dekker trains					
Detailed analysis of code	Correct implementation	Mostly correct implementation using train code	Code unlikely to be in terms of trains — mostly regurgitated from lecture	Answer has little relevance	NS
Week 15: Semaphores					
Full explanation for both cases	Provides detailed explanation for the deadlock and/or correct explanation for the <code>put</code> method case	Correctly identifies deadlock, and provides some explanation	Correctly identifies error, but possibly doesn't recognise it as deadlock	Shows little understanding of semaphores	NS
Week 16: Locks, Conditions					
Full documentation and/or testing	A correct implementation, but possibly lacking in documentation	A (near) working solution, but with some shortcomings	Some attempt to use Locks — e.g. <code>lock()</code> and <code>unlock()</code>	Mostly regurgitated code from the lecture's bounded buffer example	NS

2.2.4 Submission 4: 17.15, Tuesday 9/4/2019

Submission 4: 17.15, Tuesday 9/4/2019					
5	4	3	2	1	0
Week 17: Modelling circuits					
Full explanation of derivation of the matrix. Testing of the model and/or alternative derivation	Generally correct matrix model	An attempt to model the circuit with matrices	Little understanding of the exercise	NS	
Weeks 19–20: Quantum computing					
Shows understanding that simple probability model entails loss of information making reversability impossible	Well presented working for matrix arithmetic. Realisation that input returns to its initial value	Full set of matrices for values at A , B and C	Little more than repetition of answers to 1(a) and 1(b)	Shows little understanding of question	NS
Week 21: Correctness					
A complete proof of the algorithm, with excellent explanation	A full, or nearly full proof of the algorithm, with good explanation	A good attempt at using assertions, possibly with minor errors. Reasonable explanation	Some attempt at using assertions, but significant shortcomings, and little explanation	Shows little understanding of assertions	NS
Week 22: Complexity					
Correct implementations of both methods, with the required complexity. Excellent analysis/explanation of the complexities	Correct implementations of both methods, with the required complexity. Good analysis/explanation of the complexities	A fair attempt to implement the methods. If the required complexity is not achieved, the explanation suggests that this was a failure of implementation, rather than of understanding	Some attempt to implement the methods. While one or both of the implementations may not have the required complexity, the intention is clear, backed up by explanation	Little attempt to implement the methods	NS

2.3 Logbook

Logbook					
5	4	3	2	1	0
Presentation & Structure:					
Imaginatively presented resulting in clarity of message and information	Material is carefully structured with a clear message and visual effect	Some structural elements absent	Material presented is relevant but lacks structure or visual impact	Not all material is relevant and/or is disorganised. Lacks structure. Illegible fonts	NS
Clarity of Expression:					
Fluent writing style appropriate to the document. Grammar and spelling accurate	Language fluent. Only very minor grammar and spelling errors, if any	Language mainly fluent. Grammar and spelling mainly accurate	Meaning apparent, not always fluent. Contains grammar and/or spelling errors	Meaning unclear and/or frequent errors in grammar and/or spelling	NS
Code:					
Excellent layout, documentation. No obvious errors. Excellent use of generics	Good layout, documentation. Practically no easily avoidable errors. Good use of generics	Legible code. Reasonable layout, documentation. Few basic errors. Some use of generics	Most code legible. Some common errors (e.g. unwarranted print statements, autogenerated comments, ...)	Code poorly presented. Very little documentation. Easily avoided errors	NS
Testing:					
Excellent use of test suites	Good use of test suites	Some use of test suites	Some testing, but possibly by inspection of output	Little evidence of testing	NS

2.4 Self-assessment

Your self-assessment will be marked on its accuracy, and on how well justified it is.

2.4.1 Accuracy

If your assessment of your mark matches your actual mark, you will be awarded 5 marks for accuracy. If your assessment and your actual mark differ by one, you will be awarded 4 marks, and if they differ by two you will be awarded 2 marks. If the difference is three or more, you will gain no marks. This is shown in table below:

	0	1	2	3	4	5
0	5	4	2	0	0	0
1	4	5	4	2	0	0
2	2	4	5	4	2	0
3	0	2	4	5	4	2
4	0	0	2	4	5	4
5	0	0	0	2	4	5

2.4.2 Justification

You will also be given a mark out of five for the strength of the justification that you give for your self-assessed mark. The marks will be awarded roughly as follows:

0: No justification given

1: A very poor justification

2: Justification just acceptable

3: A reasonable justification

4: A strong justification

5: An excellent justification

2.5 Additional work

Finally, you can potentially earn some extra points by including some additional work in your logbook. This could be some of the additional exercises provided in the practical exercises, or it could be other relevant exercises that you have found elsewhere (e.g. Project Euler ⁵), or that you have thought of yourself.

Additional work					
5	4	3	2	1	0
Practical exercises: Additional work on other practical questions					
Excellent work on a comprehensive selection of the additional questions	Strong work on a good selection of the additional questions	Good attempts at a small selection of additional questions	Fair attempts at a couple of additional questions	Poor attempts at just one or two additional questions	NS
Own exercises: Work on questions in addition to the questions provided					
Excellent work on a good selection of relevant questions	Strong and clearly relevant work	Some good work on well thought out questions	Attempts at a couple of reasonably relevant questions	One or two questions of dubious relevance	NS

⁵<https://projecteuler.net/>