CAB203

Problem solving exercises

1 Introduction

In this assessment item will demonstrate your knowledge and skill with the material from the special topics lectures by developing two solutions to two real world problems using ideas and techniques developed in the special topics lectures. You will also implement your solution in Python to solve the problem.

At the end of this brief there are three topics, each describing a different problem from one of the special topics. You will choose any *two* of these problems to solve. Your job will be to use ideas from the unit to describe the problem and a solution mathematically in a report and to implement your solution in Python.

Your submission will consist of two parts: a report detailing the mathematical descriptions of your two chosen problems and solutions, and a Python file containing both of your implemented solutions.

This assignment is worth 30% of your final grade.

Due date: 22 June 2022, 11:59:00pm

2 Report

Your report should consist of two sections, one for each of your chosen topic. Each topic specifies what should go into your report. Each section is intended to be brief. To help with this, you will not write an introduction; you may assume that the reader has just read the statement of the problem in this document. While there is no enforced limit, you can expect to take about one page for each topic. You may assume that the reader is familiar with the material in CAB203.

3 Python implementation

The problems are all solvable using the Python concepts and syntax used in the unit, although you may need to do some of your own research for some specific libraries like `re` or `numpy`. Each topic specifies a function name, function parameters, and what your function should return. An automated test system will run your functions and assign a score based on the marking criteria given below in Section 6. A template Python submission file, including a test harness, will be made available on Blackboard.

4 Submission

Submission process: You will need to make two submissions:

- Your report, via Turnitin, in PDF format
- Your Python code, via Blackboard assignment, as single plaintext Python source code containing your functions for both of your chosen topics.

Links will be available on Blackboard on the Problem Solving page. You can make as many submissions as you like up to the due date, but only your most recent submission will be saved.

Extensions: Extension requests are processed by student services (`science@qut.edu.au`). There is a link on Blackboard on the Problem Solving page where you can apply for an extension. If you make a submission before the due date then after the due date further submissions will no longer be accepted by Turnitin. In this case you will need to email the unit coordinator (`matthew.mckague@qut.edu.au`) to delete your previous submission so that you can make a late submission.

48-hour extensions are available. Because of this, the due dates for Turnitin and Blackboard will be set to 48 hours after the set due date to allow you to submit late if you have an extension.

Citing your sources: You are welcome to source information and code from the internet or other sources. However, to avoid committing academic misconduct, you must cite any sources that you use.

You are welcome to use resources, including code, from within the unit. Please cite the unit like *CAB203, Tutorial 7* or similar. This is only necessary when explicitly quoting unit material. There is no need, for example, to cite the definition of a graph or similar, unless you are directly quoting the lecture's definition.

For code, please include your citation as a comment within the code. For example

```
# modified from CAB203 graphs.py
```

Policy on collaboration: We encourage you to learn from your peers. However, for assessment you need to turn in your own work, and you learn best if you have spent some time thinking about the problem for yourself before talking with others. For this reason, talking with other students about the project is encouraged, as long as you are putting in the effort on the problems yourself as well. But do not share your code or your report with other students and do not copy from others. It is considered academic misconduct to copy from other students or to provide your work to others for the purposes of copying.

For Slack and other online discussions, please do not post about solutions. Keep your discussions private so that everyone gets a chance to get to the solutions on their own.

5 Topics

Choose *two* of the topics below. Clearly indicate which ones you have chosen through appropriate section titles in your report and by implementing the corresponding functions in the Python template file.

5.1 Regular languages and finite state automata

Your unit coordinator for CAB203 has given you the absurd task of censoring outgoing student emails. You have decided to respond by censoring with an equally absurd method: redacting all articles (i.e. *a, an, the*). There is only one catch, and that is Catch-22: you need to append your student number to all emails that you censor.

Write a Python function `censor(s)` that takes string `s` and returns it back, but with all occurrences of *a, an, the* (regardless of cApitALiSatIOn) replaced with `##`, where the number of `#` characters matches the number in the word being redacted. For example `an` should be replaced with `##`. If your function makes any replacement (and only if it makes a replacement) it should append to the string a single space followed by your student number to the string in the format `<n1234567>`. For example:

```
The cat ate a mouse.
```

should become

```
### cat ate # mouse. <n1234567>
```

(with your own student number substituted!)

Your solution should take advantage of some of the richer regular expression syntax and functions available in Python, which will make it much easier! See `https://docs.python.org/3/library/re.html` for documentation. For your report, write a short introduction to the regular expressions and functions that you use in your function. You may assume that the reader has knowledge of the material from CAB203.

5.2 Linear algebra

A farmer needs to select fertiliser for her crops. She has access to two different types of fertiliser. Type A contains 30% nitrogen and 20% phosphate. Type B contains 10% nitrogen and 40% phosphate. Supposing that a particular crop needs n kilograms of nitrogen in total, and p kilograms of phosphate in total, how many kilograms of each type of fertiliser are required?

Your solution should work with different types of fertiliser with different amounts of nitrogen and phosphate (but only two types at a time) and different crop requirements.

Your implementation should be a function `fertiliser(an, ap, bn, bp, n, p)` where:

- `an` is the amount of nitrogen in 1kg of type A fertiliser
- `ap` is the amount of phosphate in 1kg of type A fertiliser
- `bn` is the amount of nitrogen in 1kg of type B fertiliser
- `bp` is the amount of phosphate in 1kg of type B fertiliser

- n is the amount of nitrogen required by the crop
- p is the amount of phosphate required by the crop

Your function should return a pair (a,b) where a is the amount of fertiliser of type A required, and b is the amount of fertiliser of type B required. All amounts of fertiliser and nutrients are measured in kg. If there is no solution to the problem your function should instead return None. Note that in this scenario, the farmer cannot add a negative amount of fertiliser to her field, so if a or b is negative then that does not count as a solution to the problem. Likewise, all the arguments to the function will make sense physically, i.e. they will all be non-negative. Mathematically, it is possible for there to be an infinite number of solutions, but this is beyond what we studied in the unit, so such cases can be ignored; they will not be used as test cases.

Your report should provide a mathematical description of how you solved the problem, including any equations that you use and your method of solving them. You may assume that the reader is familiar with the material in CAB203.

Hint: This problem can be modelled as a pair of linear equations in two unknowns.

5.3 Probability

You have found a dodgy bookmaker who plays a strange game. He has two coins, both biased. The first one comes up heads 70% of the time. The other comes up tails 60% of the time. To you, they look identical.

The bookmaker's process is to approach you with a single coin. You can't tell which one, but it is one of the ones described above. He will then state betting odds for heads and for tails and ask if you want to bet. Your options are to bet heads, bet tails, or make no bet. He then tosses the coin and pays out based on the odds if it matches your bet. The bookmaker will repeat this process 100 times *with the same coin* but offering different odds each time. The bookmaker's odds seem random and are always between 1 and 3.

Your goal is to write a Python function makeBet(headsOdds, tailsOdds, previousOutcome, state) that makes bets with the bookmaker. The marker will call your function as follows:

- headsOdds and tailsOdds give the total payout (including your stake of 1) for a bet of heads or tails respectively.
- previousOutcome gives the outcome of the *previous* coin toss, which you should use to update your models
- Your function should return a pair: (bet, state)
- bet should be one of 'heads', 'tails', 'no bet' representing your bet
- state is a variable that will be returned to you next time your function is called. I.e. if your function returns state then that will be given as the state argument on the next call to your function. You can use this mechanism to allow your function to keep track of any information that you see fit.

The marker will randomly choose one of the two coins and call your function 100 times with the same coin. The first time, state and previousOutcome will be set to None. The marker will keep track of the profit that you make over the 100 calls. Your goal is to make the most profit. A test harness will be made available that gives the details of this process.

Since this is randomised process, the profit will change from run to run. To smooth out the differences, the marker will make 10000 runs as described above, each time with a different coin (total 1000000 calls to your function, 100 in a row with each coin). Your profit will be taken to be the average profit over these 1000 runs. A profit of about 33.6 is achievable using all information, 22 is achievable without making use of the `state` variable, and 10 is achievable using a constant output.

Your report should provide a mathematical description of how you solved the problem, including any equations that you use and your method of solving them.

6 Marking criteria

Your submission is marked out of 30 marks total, distributed as detailed below.

6.1 Report

The report is out of 16 marks total. Your report will receive marks according to three criteria: presentation (4 marks) and two criteria corresponding to your two chosen topics (6 marks each).

6.1.1 Presentation

4 Marks	3 Marks	2 Marks	1 Mark	0 Marks
The overall presentation of the report is clear, including appropriate sections, with minimal grammatical or stylistic problems. Appropriate citations are given where relevant. Any equations are typeset and diagrams are created digitally.	Minor problems with presentation, sectioning, grammar or citations that don't impede the reader's understanding	Some problems with presentation such as inconsistent formatting, sections not matching those in the brief, or multiple problems with grammar or spelling. Citations are mostly present where appropriate, but using an inconsistent style or some needed citations are missing. Equations and diagrams (if present) are tidy but hand drawn rather than being created digitally.	Presentation is poor, using inappropriate or no formatting, sections are missing or inappropriate, or problems with grammar or spelling cause problems for the audience to understand. Citations are absent or inappropriate. Equations and diagrams (if present) are untidy.	Presentation is unacceptably poor for a University paper. Writing is disorganised, difficult to understand and displays a lack of basic proof reading. Citations, equations and diagrams are completely missing where they would be appropriate.

6.1.2 Regular expressions and finite state automata

6 Marks	5 Marks	4 Marks	3–2 Marks	1–0 Marks
The report gives a clear, brief description of 3 to 4 regular expression functions or syntax used in Python that are not described in the unit material and are useful in solving the problem. Explanations make reference to terms and concepts used in the unit and describe how the functions and syntax are used in solving the problem. The report is clearly understandable for someone with knowledge of CAB203.	The report gives a good description of 3 to 4 regular expression functions or syntax used in Python that are not described in the unit material and are useful in solving the problem. Explanations make some reference to terms and concepts used in the unit and describe how the functions and syntax are used in solving the problem. The report is understandable for someone with knowledge of CAB203.	The report gives a description of 1 to 2 regular expression functions or syntax used in Python, but they may not be useful to the problem, or may repeat concepts from the unit. Descriptions have some factual errors or other problems. Explanations make limited reference to terms and concepts used in the unit and only vaguely describe how the functions and syntax are used in solving the problem. The report is difficult to understand for someone with knowledge of CAB203.	The report gives a description of at least 1 regular expression functions or syntax used in Python, but the description has significant problems or errors or is not relevant to the problem. The explanation has a minimal relationship to CAB203 material.	The report does not give a recognisable description of regular expression functions or syntax used in Python. The explanation has no relationship to CAB203 material or the problem.

6.1.3 Linear algebra

6 Marks	5 Marks	4 Marks	3–2 Marks	1–0 Marks
The report gives a clear mathematical description of the method used to solve the problem in terms of linear algebra. Mathematical terms and notation are used accurately and appropriately. The level of explanation is appropriate and understandable for someone with knowledge of CAB203 material.	The report gives a mathematical description of the method used to solve the problem in terms of linear algebra and with minor errors only. Mathematical terms and notation are mostly used accurately and appropriately with minor problems only. The level of explanation is appropriate and understandable for someone with knowledge of CAB203 material.	The report gives a description of the method used to solve the problem, but it has errors and makes little reference to linear algebra. Mathematical terms and notation are used, but with some significant errors. The explanation is difficult to understand for someone with knowledge of CAB203 material.	The report gives a description of the method used to solve the problem, but it is informal and does not make significant use of mathematical concepts. Usage of mathematical terms and notation is infrequent and problematic. The explanation has a minimal relationship to CAB203 material.	The report's description of the method used is problematic and mostly erroneous. Mathematical terms and notation are absent or used completely inappropriately. The explanation does not relate to CAB203 material.

6.1.4 Probability

6 Marks	5 Marks	4 Marks	3–2 Marks	1–0 Marks
The report gives a clear mathematical description of the method used to solve the problem in terms of probabilities. Mathematical terms and notation are used accurately and appropriately. The level of explanation is appropriate and understandable for someone with knowledge of CAB203 material.	The report gives a mathematical description of the method used to solve the problem in terms of probabilities and with minor errors only. Mathematical terms and notation are mostly used accurately and appropriately with minor problems only. The level of explanation is appropriate and understandable for someone with knowledge of CAB203 material.	The report gives a description of the method used to solve the problem, but it has errors and makes little reference to probabilities. Mathematical terms and notation are used, but with some significant errors. The explanation is difficult to understand for someone with knowledge of CAB203 material.	The report gives a description of the method used to solve the problem, but it is informal and does not make significant use of mathematical concepts. Usage of mathematical terms and notation is infrequent and problematic. The explanation has a minimal relationship to CAB203 material.	The report's description of the method used is problematic and mostly erroneous. Mathematical terms and notation are absent or used completely inappropriately. The explanation does not relate to CAB203 material.

6.2 Implementation

The automated marking system will run your solution against test cases for your chosen topics. Please note that these test cases will not be made available, other than the case given with the test harness for each topic. You will receive marks for each of your two chosen topics to a maximum of 7 for each topic, 14 in total.

- **Regular language and finite state automata:** Your implementation will be tested against 7 test cases designed to test various aspects of the problem. You will receive one mark for each test case where your function's output matches the specification.
- **Linear algebra:** Your implementation will be tested against 7 test cases designed to test various aspects of the problem. You will receive one mark for each test case where your function's output matches the specification.

- **Probability:** Your implementation will be tested exactly as in the test harness given, but with a different random seed. You will be given[1]

$$\left\lceil \frac{7p}{33.5} \right\rceil$$

marks, where p is the average profit per run, as reported by the test harness. 7 is full marks, but if you achieve more than 7 according to the formula, the excess are treated as bonus marks.

[1] You may wish to review the material in week 1 if the notation used looks unfamiliar.