COS4015-B Technical and Professional Skills (TPS)

Coursework 1

Algorithms & Programming (Team Submission)

Summative Assessment: 50% weighting; team-based development of a prototype

In this assessment, you are expected to write an algorithm for a particular problem and implement your solution in **Python**.

You are given a list of computational problems on page 3. Select ONLY ONE and write an algorithm that solves the problem you have picked. You are strongly encouraged to produce **general solutions** rather than specific solutions. Your submission should include (in a **SINGLE** file):

1. **Pseudocode** that describes the steps of your algorithm. You can write your solution in any text editor. You need to follow conventional Pseudocode style(s), as presented in the lecture.
2. **Flowchart** that visually represents your algorithm. You can use any tool to draw your diagrams. Make sure that you export your drawings in a common format, e.g., pdf, jpeg, doc, etc.
3. **Computational complexity** analysis of your algorithm. You can do this by either calculating the number of operations or the number of runs (in Big-Oh notation), similar to the tutorial exercises on complexity.
4. **Program** that implements your algorithm (and evidence of some test cases that your programme is actually running). You must implement your solution in **Python**. Your code will be judged on organisation, clarity, and comments/documentation.

   Compare your code with the one generated by ChatGPT (or any other large language models). Explain the differences in the algorithms used and clarify how your approach differs from ChatGPT. Attach both your code and the ChatGPT-generated code in your report.
5. **Evaluation form** summarising the contribution of each team member. If a team member does not contribute to the work, **a mark of 0** will be given. If the contribution is not satisfactory, then a **partial mark** will be given.

Deadline: 15th December @ 4pm.

Learning Outcomes: LO2, LO3, LO5, PLO3, PLO6

How and When to submit:

- Pick up 3 problems in order of your preference and email it to the module coordinator by **24th November** (Only one team representative will do that; multiple emails from the same group will be ignored). Each problem can be allocated to up to **4-5** teams, based on first come first serve basis.
- Save **pseudocode**, **flowchart**, **complexity**, and **team evaluation** as A **SINGLE** file and name it ***Team_number.pdf***. Save your Python implementation into a SINGLE file and name it ***Team_number.py***. Make sure your Python file runs in Jupyter notebook.
- You **MUST** submit your assignment to this module in Canvas (via the Assignments link) by **15th December @ 4pm**.
- Any documents submitted late will be provided feedback and **will receive a mark of 0**. The submissions that do not follow the guidelines will also be penalised.
- Work handed in on time will be marked and returned to you in 15 working days.

Marking Scheme: See attached form.

Marking Breakdown:

- **Flow charts (20 marks):** marks will be deducted in case of using wrong flow chart symbols and/or providing a chart which is different than your pseudocode and implementation.
- **Pseudocode (20 marks):** marks will be deducted in case of using wrong/unconventional pseudocode format and/or providing a pseudocode which is different than your flowchart and implementation.
- **Implementation (40 marks):** marks will be deducted in case of programming related issues, lack of organisation, clarity, comments and/or providing a prototype/implementation which is different than your flowchart and pseudocode. Also, more general solutions will get higher marks.
- **Complexity (20 marks):** marks will be deducted if complexity calculations have errors and/or there is not sufficient explanation / justification of your analysis.

Problems

Problem 1: A student finds herself in a riverbank with three pets: a dog, a cat, and a hamster. She needs to transport all three to the other side of the river in her boat. However, the boat has room for only the student herself and one pet (either the dog, the cat, or the hamster). In her absence, the dog would attack the cat, and the cat would attack the hamster. Devise an algorithm that shows how the student can get all these passengers to the other side.

Problem 2: The first 50 numbers – 1,2,...,50 – are written on a board. You have to apply the following operation 49 times; select two of the number on the board, *a* and *b*, write the absolute value of their difference |a — b| on the board, and then erase both *a* and *b*. Write an algorithm that determines the remaining number that can be obtained in this manner.

Problem 3: The *n* consecutive integers from 1 to *n* are written in a row. Design an algorithm that puts signs '+' and '—' in front of them so that the expression obtained is equal to 0 or, if the task is impossible to do, returns the message no solution.

Problem 4: You have to make $n \geq 1$ pancakes using a skillet that can hold only two pancakes at a time. Each pancake has to be fried on both sides; frying one side of a pancake takes 1 minute, regardless of whether one or two pancakes are fried at the same time. Design an algorithm to do this job in the minimum amount of time.

Problem 5: A nursery teacher has to arrange $2n$ children in *n* pairs for daily walks. Design an algorithm for this task so that for $2n - 1$ days no pair would be the same.

Problem 6: There are 20 gloves in a drawer: 5 pairs of black gloves, 3 pairs of brown, and 2 pairs of grey. You select the gloves in the dark and can check them only after a selection has been made. Write an algorithm that finds the smallest number of gloves you need to select to guarantee getting at least one matching pair of each colour.

Problem 7: You have 20 black balls and 16 white balls in a bag. You repeat the following operation until a single ball is left in the bag. You remove two balls at a time. If they are of the same color, you add a black ball to the bag; if they are of different colors, you add a white ball to the bag. Outline an algorithm that predicts the color of the last ball left in the bag.

Problem 8: There are *n* index cards in a row, with *n* distinct integers written on them (one number per card) so that the numbers are sorted in decreasing order. You are allowed to exchange any pair of cards that have exactly one card between them. Write an algorithm that sorts the cards in increasing order with the minimum number of exchanges.