UNIVERSITY OF GREENWICH

COMP1819 **Algorithms and Data Structures**	2024/25	**Coursework ID:**	**Contribution: 100% of course**
Coordinator: **Dr Tuan Vuong**	**Release Date:** 03/02/2024	**Deadline Date:** 18/03/2025	**Return Date:** 09/04/2025
This coursework should take an average student who is up-to-date with tutorial work for approximately 40 hours. Feedback and grades are normally available within 15 working days of the coursework deadline.			
Learning Outcomes: 1. Select and employ data structures appropriate to various problems. 2. Understand the relationship between algorithms and data structures. 3. Formulate and solve programming problems using learned concepts. 4. Understand the complexity of algorithms in terms of time and memory.			

Coursework Submission Requirements

- An electronic copy of your work for this coursework must be fully uploaded on the Deadline of **Tuesday 18/03/2025** using the link on the coursework Moodle page for COMP1819.
- For this coursework you must submit a single PDF document. In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand-written mathematical notation, but when scanning do ensure the file size is not excessive.
- For this coursework you must also upload a single **ZIP** file containing supporting evidence.
- There are limits on the file size (see the relevant course Moodle page).
- Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.
- Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.
- You must NOT submit a paper copy of this coursework.
- All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See http://www2.gre.ac.uk/current-students/regs

Algorithms and Data Structures (ADS) - COMP1819

Develop and optimise solutions in Python with ADS and provide complexity analysis.

This is a GROUP coursework.

Detailed Specification

Problem Title: Unique Prime Numbers "Hidden" in Binary Strings

Your group has been tasked with solving a coding interview question in Python designed to assess a candidate's understanding of **fundamental data structures and algorithms** at various levels of optimisation. This challenge will help you develop key skills, including selecting and implementing appropriate **data structures**, designing **efficient algorithms**, analysing **computational complexity**, and optimising solutions to handle **large inputs** effectively.

Your task is to write a Python program that finds **all unique prime numbers** hidden within a given **binary string, and less than a given integer number N**.

For example, given the binary string **"0110" and the number 5**, we can extract multiple substrings, such as **"0"**, **"1"**, **"11"**, **"10"**, **"110"…** Converting these binary substrings into their decimal equivalents gives the numbers **0, 1, 2, 3, 6,** and so on. Among these, we identify the **prime numbers** that are **less than a given value 5**. In this case, the prime numbers are **2** (decimal equivalent of binary string "10") **and 3** (decimal equivalent of binary string "11") after removing duplicates. So the output is "2: 2, 3"

We need to print all the prime numbers, if **fewer than 6 prime numbers** are found. However, if **6 or more** prime numbers are found, it should display **only the first three smallest primes and the last three largest ones**.

Examples:

Input	Output (Total: Primes Numbers)	Explaining
101011,99	5: 2, 3, 5, 11, 43	Total prime numbers = 5 List of prime numbers = 2, 3, 5, 11, 43
0110,5	2: 2, 3	Total prime numbers = 2 List of prime numbers = 2,3
COMP,4	0: Invalid binary strings	Not binary strings

The 10 Given Test cases:

#	Input (extra space for readability)	Output	Comment (Extra Information for your knowledge)
1	0100001101001111,999999	Your answers?	The binary representation of 'CO'
2	01000011010011110100110101010000,999999	Your answers?	The binary representation of 'COMP'
3	1111111111111111111111111111111111111111, 999999	Your answers?	40 1's
4	010000110100111101001101010100000011000100111000,999999999	Your answers?	The binary representation of 'COMP18'
5	01000011010011110100110101010000001100010011100000110001,123456789012	Your answers?	The binary representation of 'COMP181'
6	0100001101001111010011010101000000110001001110000011000100111001,123456789012345	Your answers?	The binary representation of 'COMP1819'
7	010000110100111101001101010100000011000100111000001100010011100100100001,123456789012345678	Your answers?	The binary representation of 'COMP1819!'
8	01000011010011110100110101010000001100010011100000110001001110010010000101000001,1234567890123456789	Your answers?	The binary representation of 'COMP1819!A'
9	0100001101001111010011010101000000110001001110000011000100111001001000010100000101000100,1234567890123456789	Your answers?	The binary representation of 'COMP1819!AD'
10	010000110100111101001101010100000011000100111000001100010011100100100001010000010100010001010011,12345678901234567890	Your answers?	The binary representation of 'COMP1819!ADS'

Your group must develop multiple solutions for the given problem, providing complexity analysis, test cases, results, and the corresponding Python code. The program must not rely on a hard-coded list of values, except for up to five predefined values, and external data files are not allowed. It should aim to complete execution within 60 seconds for any test case. Solutions must be your code, though you may use Python's built-in libraries. Additionally, the program should be concise, ideally not exceeding 100 lines of code.

Deliverables – Submission requirements

Deliverable 1 – the report in PDF format

A template will be provided. **Your report must be created using Online Document Sharing** such as Microsoft 365 for ongoing collaboration with the tracker on. You must provide a link to this document (make it viewable to everyone with the link) in the final report you submit. Each team member must select different colours for their respective text and provide a legend of which colour text belongs to which team member at the begin of the report.

1. Creating two basic working solutions (30%) – Maximum 2 pages

- Each team member must explain their understanding of the problem with clearly **handwritten examples (at least 5 in total) with different input string lengths and their expected outputs. This must be later used to validate the correctness of your solutions.**
- Implement **two distinct solutions**, each using a **different data structure**.
- Include your **full code** in the **Appendix of the report (as clear text, not a screenshot)**.

2. Testing and Compare Your Solution (20%) – Maximum 2 pages

- Test your solutions against the test cases (examples) you used in in Task 1 in a **clear table with evidence** (snapshot of output generated from your implementation.)
- Run your solutions on the **10 given test cases.** Record **the output and the running time** for each (you can stop after 60 seconds) in the report. Also put these details as comment in your code.
- Compare your two solutions, highlighting the differences in **data structures** and their impact on performance.
- Provide a theoretical analysis of the complexity of your approaches and generate a **graph** showing how the runtime of both solutions changes across the **10 given test inputs**.

3. Optimising Solutions (20%) – Maximum 2 pages

- Choose **one basic solution** to optimise and improve its efficiency to solve **as many given test cases as possible** within a **60-second runtime limit**.
- Explain each **optimisation step**, detailing how it enhances performance (**Miller-Rabin method and external libraries are not allowed**). Your optimisation should be an improvement of the basic solution (one of the distinct solutions you obtained in Task-1 and not completely new and different).
- Include the **final optimised code** in the **Appendix of the report (as clear text, not a screenshot)**.

4. Comparing Performance (15%) – Maximum 2 pages

- Compare the optimised solution with the basic solution focusing on differences in **data structure, algorithm, or implementation approach**.
- Run your **optimised solution** on the **10 given test cases.** Record **the output and the running time** for each (you can stop after 60 seconds) in the report. Also put these details as comment in your code.
- Create a **graph** to visually compare **runtime performance** across different test cases between the optimised solution **and** the basic solution.
- Analyse and explain the **time complexity** and **Big-O notation** for the optimised solution.

5. Reflecting on Teamwork (15%) – Maximum 2 pages

- Assign **contribution marks** to each team member based on mutual agreement.
- Maintain a **weekly journal** tracking communication logs and individual contributions.
- Provide a **clear summary** of each member's role and specific contributions to the project.
- Ensure the **final report** is **well-organised, clearly written, properly referenced**, and follows the given specifications.

Deliverable 2 – Source code & test case input files (marks included within Deliverable 1.1, 1.3)

The Python source codes (one for each solution), inputs/test cases used for analysis, and others. They should be placed in a zip file which must be uploaded separately from the report.

You must implement your solution in **Python** programming language. You may use any sample code provided in the course lectures and laboratories/tutorials as an aid, but make sure you reference any source code or tools (including AI tools) used.

Unreferenced codes/contents may involve an investigation into **an academic misconduct offence.**

You are strongly advised to commence working on this coursework **as it becomes available** and contact the **Module leader** with your query as early as possible.

Additional coursework guidelines and advice will be provided separately.

Grading Criteria

Criteria for Assessment	80-100	70-79	60-69	50-59	40-49	30-39	0-29
Content, knowledge and understanding	Demonstrates exceptional systematic understanding of problem solving, computer programming and algorithmic performance. There is exceptional evidence of engagement with all key elements.	Demonstrates an excellent systematic understanding of problem solving, computer programming and algorithmic performance. There is also excellent evidence of engagement with all key elements.	There is a very good systematic understanding of problem solving, computer programming and algorithmic performance. There is also some very good evidence of engagement with all key elements.	Has demonstrated a good understanding problem solving, computer programming and algorithmic performance. There is also some good evidence of engagement with most key elements with some omission of detail.	Has demonstrated a satisfactory level of understanding of problem solving, computer programming and algorithmic performance. There are a few notable omissions and there is limited evidence of engagement with all key elements. Overall a satisfactory attempt at these criteria	A poor understanding of one or more of the following - problem solving, computer programming and algorithmic performance. There is insufficient evidence of engagement with the key elements. Overall an unsatisfactory attempt.	Little or no understanding of one or more of the following - problem solving, computer programming and algorithmic performance. There is very little evidence of engagement with the key elements. Overall a very unsatisfactory attempt.
Cognitive/Intellectual Skills	Demonstrates exceptional use of a critical analysis of information leading to the proposal of a robust and detailed solution. There is exceptional evidence of reflection that identifies the strengths and weakness of the approaches undertaken.	Demonstrates an excellent use of a critical analysis of information leading to the proposal of a robust and detailed solution. There is also excellent evidence of reflection and judgement based on the interpretation of the results obtained.	Demonstrates a very good use of a critical analysis of information leading to the proposal of a detailed solution. There is also some very good evidence of reflection and judgement based on the interpretation of the results obtained.	Demonstrates some good critical analysis of information leading to the proposal of a detailed solution. There are some exposed weaknesses of cognitive skills. There is also some good evidence of reflection and judgement based on the interpretation of the results obtained.	Has shown some satisfactory level of critical analysis of information. There is evidence of reflection and judgement based on the interpretation of the results obtained at a threshold pass level.	Has shown little use of techniques to undertake a critical analysis of information. The reflection and judgement based on the interpretation of results is weak and lacks detail.	Has shown little or no use of techniques to undertake a critical analysis of information. The reflection and judgement based on the interpretation of results is very weak and lacks detail.

Communication, Organisation and Presentation **Graduate Employability and Application of Skills**	Demonstrates exceptional use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and of an exceptional quality. There is exceptional evidence of the qualities of transferrable skills necessary for employment that required personal judgement and successful experimentation.	Demonstrates excellent use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and of an excellent quality. There is excellent evidence of the qualities of transferrable skills necessary for employment that required personal judgement and successful experimentation.	Demonstrates a very good use of argument and language which effectively communicates information to the target audience. The structure and flow of the report is clear and overall is very good. There is also very good evidence of the qualities of transferrable skills necessary for employment that required personal judgement and mostly successful experimentation.	There is good use of argument and language which communicates information to the target audience. The structure and flow of the report is mostly coherent and overall is good. There is also some good evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is mostly acceptable with some shortcomings in the grammar. The structure and flow of the report is barely acceptable with some presentation issues. There is also some evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is mostly at a substandard level. The structure and flow of the report is unacceptable with some presentation issues. There may also be little evidence of the qualities of transferrable skills necessary for employment.	The use of argument and language which communicates information to the target audience is at a substandard level. The structure and flow of the report is unacceptable with significant presentation issues. There may also be little/no evidence of the qualities of transferrable skills necessary for employment.
Referencing, sourcing, acknowledging and coverage	The exceptional use of appropriate references reflects clear and detailed understanding of the referenced works and its contents from a variety of sources.	The excellent use of appropriate references reflects clear and detailed understanding of the referenced works and its contents referenced works.	The use of references reflects a very good understanding of the cited work and its contents. Some references may not be the most recent.	The use of references reflects a good understanding of the cited work and its contents. Some references may not be the most recent or are taken from a narrow range of sources.	The use of references reflects a satisfactory understanding of the cited work and its contents. Some references may not be appropriate or the most recent or are taken from a narrow range of sources.	The use of references reflects a poor understanding of the cited work and its contents. The references may not be sufficient or appropriate or the most recent or are taken from a narrow range of sources.	Little or no cited work. The references may not be appropriate or the most recent.