PG4200: Algorithms and Data Structures
Prof. Dr. Rashmi Gupta
Kristiania University College, Oslo, Norway

Final Exam

- This exam contains six major questions, some with multiple parts. You have 24 hours to earn 100 points.
- This exam is an open book. You may use handwritten A4 sheets, MS Word documents for theoretical answers, and IntelliJ IDE for Java code implementation. The use of calculators or programmable devices is permitted.
- Please ensure your handwriting is **legible** (I cannot grade that I cannot read!). Before you upload your solutions, ensure all your answers are included and matched numerically with your question paper.
- Start uploading your exam paper/file ahead of time, as it may take a long time to upload.
- Exam papers that are not handed in on Wiseflow by the specified time of the submission date will not be proceeded to assessment. No late solutions will be accepted.
- Before submitting, remember to check that all files can be opened and that every file is included. It may be a good idea to check the saved files on several machines before submitting them in Wiseflow.
- Incorrect file format or lacking documents may result in the submission not being passed or assessed.
- The total file size of the entire exam paper/file must not exceed 5GB in zip format or 5MB in PDF format.
- Do not spend too much time on any one Question. Read them through first, and start solving them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer but also on the clarity with which you express it.
- The use of ChatGPT is strictly prohibited. Your submitted documents will be checked for plagiarism.

Good luck!

Questions	Title	Type of Explanation	Points	Score
1	LO1: Understanding Data Structures	Algorithmic	15	
2	LO1: Understanding Data Structures	Algorithmic/code implementation	15	
3	LO2: Searching Algorithms	Algorithmic/graphical	15	
4	LO 3: Sorting Algorithms	Algorithmic/graphical	15	
5	LO 4: Traversing Graphs Algorithms	Algorithmic/code/graphical	20	
6	LO 5 and LO 6: Computability and Complexity	short answers	20	
Total			**100**	

Question 1 (15 points): LO1: Understanding Data Structures

- **Question 1.1 (7.5 points, Formula and Calculation):** How to access any random element from a 1-D array? Where the given base address of an array **A[1300 ………… 1900]** is **1022**, and the size of each element is **2 bytes** in the memory, find the address of **A[1704].**
- **Question 1.2 (7.5 points, Algorithm):** Write a step-by-step algorithm that traverses the whole array and finds the largest number among them.

Question 2 (15 points): LO1: Stack (push/pop/getMin operations)

- **Question 2.1 (7.5 points: Algorithms):** Write the algorithms for Push and Pop operations in the stack data structure.

- **Question 2.2 (7.5 points: Code and Explanation):** You are given **N** elements , and your task is to Implement a Stack in which you can get a minimum element in O(1) time. You are required to complete the three methods:
  - **push()** which takes one argument, an integer **'x'** to be pushed into the stack,
  - **pop()** which returns an integer popped out from the stack, and
  - **getMin()** which returns the min element from the stack. (-1 will be returned if for **pop() and getMin()** the stack is empty.)
  - **Constraints:**
    1 <= Number of queries <= 100
    1 <= values of the stack <= 100

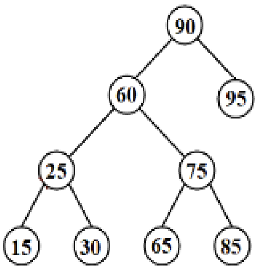
Question 3 (15 points): LO2: Searching Algorithms

- **Question 3.1 (7.5 points: Algorithms and step-by-step human-represented solution):** Write an algorithm and solution to Binary Search with a Recursive version

  Suppose *x* = 18, and we have the following array:

  10 12 13 14 18 20 25 27 30 35 40 45 47.

- **Question 3.2 (7.5 points: Algorithms and step-by-step graphical solution):** Delete **node 60** in this Binary Search Tree. Provide step-by-step algorithm and human representation way of solving the problem.



Question 4 (15 points): LO 3: Sorting Algorithms

- **Question 4.1 (7.5 points: Algorithm and Graphical Representation):** Sort the following list with **Merge Sort** and provide an algorithm and a graphical representation of solving the given problem:

  Array = *{70, 50, 30, 10, 20, 40, 60}*

- **Question 4.2 (7.5 points: Algorithm and Graphical Representation):** Sort the following list with **Quick Sort** and provide an algorithm and a graphical representation of solving the given problem:

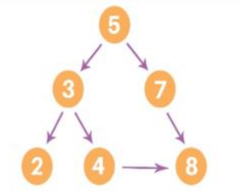
  Array = *{15, 22, 13, 27, 12, 10, 20, 25}*

Question 5 (20 points): LO 4: Traversing Graphs Algorithms

- **Question 5.1 (10 points: Code):** Regardless of how fast computers become or how cheap memory gets, efficiency will always remain an important consideration. To show the importance of developing efficient algorithms, compare these two algorithms:
    - Recursive *algorithm to solve fib*(*n*) for 1 ≤ *n* ≤ 5 Fibonacci term
    - Iterative *algorithm to solve fib*(*n*) for 1 ≤ *n* ≤ 5 Fibonacci term

    Justify which algorithm is more efficient and why through the solution with code.

- **Question 5.2 (10 points: Algorithm and human representation):** Traverse the given graph with Depth First Search and Breadth First Search algorithms:



Question 6 (20 points): LO 5 and LO 6: Computability and Complexity

- **Question 6.1 (4 points: Short Answer):** What is Complexity Class? Why is it known as complexity classes in data structures?
- **Question 6.2 (4 points: Short Answer):** Why do we use Big O Notation to analyse algorithms' complexities in data structures?
- **Question 6.3 (4 points: Short Answer):** What is P class, and What are the features of this class? Provide a real-life example of this type of problem.
- **Question 6.4 (4 points: Short Answer):** What is NP class, and What are the features of this class? Provide a real-life example of this type of problem.
- **Question 6.5 (4 points: Short Answer):** What is an NP-complete class, and What are the features of this class? Provide a real-life example of this type of problem.