PG4200: Data Structures and Algorithms

Prof. Dr. Rashmi Gupta

Kristiania University College, Oslo, Norway

Kristiania University College

Final Exam

- This exam includes a semester assignment in a group (max. 3 students).
- This means that you are being provided with an assignment and you are expected to submit below mentioned documentation in Zip file format:
  - **Assignment code scripts**
  - **Report** that explains:
    - your solutions in an algorithmic way with the relevant code snippet. Carefully explain your solutions to the problems in the text as well.
- **Duration:** 4 weeks.
- **Grading scale:** The Norwegian grading system uses the graded scale A - F, where A is the best grade, E is the lowest pass grade and F is fail
- **Weighting:** 100% of the overall grade
- 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 proceed to assessment. No late solutions will be accepted. Your submitted documents will be checked for plagiarism.
- 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.
- Show your work, and be creative, 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.

Attachment: Grading Scale

The table below shows how the Universities Norway council (UHR) defines general, qualitative requirements for the corresponding letter grades. The criteria below should be used as a guideline in assigning grades for examinations unless other assessment criteria are explicitly provided for a particular assignment.

Letter grade	Descriptor	General, not subject-specific, description of assessment criteria
A	Outstanding	**Outstanding performance that clearly stands out.** The student has extremely good knowledge of the usage of different Data Structures based on a specific given problem.
B	Very good	**Very strong performance.** The candidate displays good knowledge of learning outcomes.
C	Good	**A good performance that satisfies most assessment criteria.** The candidate displays a good knowledge in the most important areas of assessment.
D	Fair	**An acceptable performance with some clear deficiencies**. The candidate somewhat displays a level of sound knowledge of learnt DSA concepts.
E	Sufficient	**A performance that only satisfies the minimum requirements.** The candidate displays a poor level of knowledge of learnt DSA concepts.
F	Fail	**A performance that does not satisfy the most basic formal requirements.** The candidate lacks sound knowledge of learnt DSA concepts.

Point total and grading scale [1]

Letter Grade	Corresponding numerical grade	Westerdals Grading scale
A	90-100	93 -100
B	80 – 89	78 – 92
C	60 - 79	59 - 77
D	50 - 59	51 - 58
E	40 - 49	40 - 50
F	0 - 39	0 - 39

Good luck!

In this Practice, you will use the Basic World Cities Database. It can be downloaded for free from https://simplemaps.com/data/world-cities

Problem 1. BubbleSort (25 Marks)

At first, use the unique latitude values of each city only.

a. Implement an optimised and non-optimised bubble sort algorithm so that all city latitudes are in an ordered list.
b. Calculate the time complexity to sort the dataset. Does it change if you randomly order the list before sorting? Why/why not?

Problem 2. InsertionSort (25 Marks)

At first, use the unique latitude values of each city only.

a. Implement a proper insertion sort algorithm so that all city latitudes are in an ordered list.
b. Count the time complexity to sort the dataset. Does it change if you randomly order the list before sorting? Why/why not?

Problem 3. MergeSort (25 Marks)

At first, use the unique latitude values of each city only.

c. Implement a proper merge sort algorithm so that all city latitudes are in an ordered list.
d. Count the number of merges needed to sort the dataset. Does it change if you randomly order the list before sorting? Why/why not?

Problem 4: QuickSort (25 Marks)

At first, use the unique latitude values of each city only.

a. Implement a proper quick sort algorithm with following pivot selection strategies so that all city latitudes are in an ordered list.
    a. Choose **first** element as Pivot element
    b. Choose **last** element as Pivot element
    c. Choose **random** element as Pivot element
b. Count the number of comparisons for each pivot selection strategy, needed to sort the dataset. Does it change with the different pivot selection strategy? What is the best pivot selection strategy according to your given dataset?