Term Project

Total 100 points

1 Objectives

The goal of the term project is to explore an advanced topic in algorithms and data structures and investigate one or more algorithms beyond the scope of this course. You are expected to complete the following:

(1) **Understanding**: Gain a deep understanding of the selected algorithm(s) and their applications.
(2) **Pseudocode**: Describe the algorithm(s) using pseudocode.
(3) **Complexity Analysis**: Analyze the time and space complexity of each algorithm.
(4) **Implementation**: Implement the algorithm(s) in a programming language of your choice.
(5) **Experimentation**: Conduct experiments on real-world or simulated datasets to obtain meaningful results.

Please make sure that the topic(s) you select are not entirely covered in this course.

2 Important Dates

Deadline	Task
Oct 16th (11:59 PM, Wednesday)	**Finalize team members and project topic by Email** You can work individually or in groups of up to three students. All team members must be students enrolled in this course. Please email me your team members and project topic.
Oct 20th (11:59 PM, Sunday)	**Submit a project proposal via Folio** Each team will submit a project proposal in PDF format. Refer to Section 5 for specific project proposal requirements.
Dec 1st (11:59 PM, Sunday)	**Submit project paper and presentation slides via Folio** Each team will submit their latest version of project paper in PDF format with a `.zip` file containing all relevant source code. Additionally, submit one file of your presentation slides. Refer to Section 6 for detailed project paper requirements.
Dec 2nd and 4th (class time)	**Project presentation via Zoom** Each team will give a 20-30 minutes presentation discussing your project during the last class meeting day.
Dec 8th (11:59 PM, Sunday)	**Submit final project paper with source code via Folio** Submit final version of your project paper along with all relevant source files. Refer to Section 6 for detailed project paper requirements.

3 Grading

- **Project proposal** – 10 points
- **Project paper with code** – 60 points
  Note: Only your final submission before 11:59 PM, Sunday, Dec 8th will be graded.
- **Project difficulty and originality** – 10 points
- **Presentation** (slides + presentation) – 20 points

4 Possible Project Types and Topics

The expected projects may fall into the following three categories.

(1) **Reading Project**: Explore research articles on an algorithm or a class of algorithms that you find interesting. Your project will focus on reading existing works related to these algorithms. Your project paper will be like a survey that summarizes in a novel way the related research and application areas of these algorithms. You will also apply your learning by selecting at least one algorithm and conduct some experiments using a programming language of your choice.

(2) **Application Project**: Look for a real-world problem that you find interesting and explore different algorithms that can be applied to solve it. Select one or more algorithms and evaluate their performance by conducting experiments using simulated or real-world datasets. If you opt for a single algorithm, you may implement it in different ways, for example, using different data structures or using different data types as input. If you choose to apply multiple algorithms, you may compare them to see whether they would perform differently on the same input for solving the same problem. Furthermore, you can also enhance or modify the algorithms, or even design your own algorithm. **Note**: The difficulty and originality level would be higher for a project proposing an algorithm with your enhancements or original design.

(3) **Theoretical Project**: Find an open research question in algorithms. You may design new algorithms for an already studied problem or modify existing ones to tackle novel challenges. Alternatively, you can focus on enhancing the analysis of a known algorithm or demonstrating its optimality for specific inputs. You can also prove that an existing algorithm is already the best possible for certain problem scenarios. **Note**: The difficulty and originality level of a theoretical project is usually the highest. The experiments part can be skipped for a purely theoretical project.

Here are some example topics for a **Reading Project** or an **Application Project**:

- Knapsack Problem Variants and Algorithms for Solving Them
- Image Segmentation Algorithms and Applications
- Job-Shop Scheduling Algorithms and Applications
- Online Resource Allocation Algorithms and Applications
- Shortest Path Algorithms for Public Transportation
- Pattern Matching Algorithms for Efficient Search Engines
- Comparative Analysis of Data Compression Algorithms
- Graph Algorithms for Analyzing Biological Networks

If you encounter difficulties in choosing a project topic, consider the recommended course textbook and explore academic papers via Google Scholar or our University Libraries. Look for specific article titles that align with your interests and objectives, helping you narrow down your focus.

5 Project Proposal Requirements

Format: Please use Times New Roman (12 pt) or Calibri (11 pt) font, maintain single spacing, and set one-inch margins. It must not exceed 2 pages and should include:

(1) Project title
(2) Name and email address of each team member
(3) Motivation and project description
(4) Project type and timeline (with major tasks you need to complete for the project)
(5) Project goals and expected outcomes
(6) At least three references (to recent books or academic articles you plan to use)

6 Project Paper Requirements

Format: Please use Times New Roman (12 pt) or Calibri (11 pt) font, maintain single spacing, and set one-inch margins. Your paper should span between 8 and 12 pages, including the following components. Organize your paper into sections according to these components and provide a title for each section.

(1) **Title**

(2) **Team Member Information**: List the names and email addresses of all team members.

(3) **Abstract (≤ 300 words)**: Give a concise overview of your project's scope and its contributions to the field.

(4) **Introduction**: Introduce the problem and algorithm(s) under investigation. Explain why they are intriguing and worth the work you are doing.

(5) **Background and related work**: Summarize existing research and previous works relevant to your project's topic.

(6) **Algorithm(s)**: Provide a comprehensive description of the algorithm(s) you are exploring or proposing. This section should also include pseudocode for each algorithm and analysis of their time and space complexities.

(7) **Experiments**: Explain the objectives of your experiments. Write details on your approach of testing and evaluating the algorithm(s), using real-world or simulated datasets. Incorporate visual aids such as figures, plots, or tables to enhance the presentation of your results.

(8) **Conclusion**: Summarize the contributions and key insights derived from your project. Reflect on lessons learned through this process and propose potential avenues for future research or improvements.

(9) **References** (NOT counted in 12-page limit): List all the references you have used in your project paper. If you've incorporated external code, you should also provide the source of the code, such as URLs of the original source code.

(10) **Appendices** (NOT counted in 12-page limit): You may add any supplementary materials (such as more experiment results) to better support your write-up.

(11) **Contributions by Team Members** (only applicable for teams of at least 2 students): Add an additional page to list the contributions that each team member made to this project.