CALIFORNIA STATE UNIVERSITY
FULLERTON

CPSC 335: Algorithm Engineering (Spring 2024)

Department of Computer Science, College of Engineering and Computer Science

Instructor: Dr. Shah, Adjunct Faculty, Computer Science https://www.hassanonline.us/

Wi-Fi Connectivity Product Management, Qualcomm Inc., San Diego.

Project: Comparative Analysis and Visualization of Sorting Algorithms in Python

Objective

This project aims to deepen students' understanding of sorting algorithms by implementing, analyzing, and visualizing the performance of various sorting techniques in Python. Students will gain practical experience in algorithm implementation, performance analysis, and the use of visual aids to demonstrate complex concepts.

Timeframe

- **Duration:** 1 week
- **Group Size:** 4 students per group
- **Submission Due Date:** March 1, 2024 (10 am)
- **Presentation Time:** Maximum of 8 minutes, including a live demo

Roles

1. **Algorithm Implementers (2 Students):** Focus on coding the sorting algorithms.
2. **Visualization and GUI Developer (1 Student):** Develops the tool for visualizing the sorting process.
3. **Analyst and Presenter (1 Student):** Conducts performance analysis and prepares the presentation.

Project Instructions

1. **Group Formation:** Form groups of 4, assigning roles based on interests and strengths. Consider diversifying skills within each group to cover all aspects of the project efficiently.
2. **Algorithm Implementation:**
    - Implement three sorting algorithms in Python: Bubble Sort, Merge Sort, and Quick Sort.
    - Ensure your code is well-commented to explain your logic and the theoretical complexity of each algorithm.
3. **Visualization Tool Development:**
    - Utilize **matplotlib** or **Pygame** to create a visualization tool that demonstrates the sorting algorithms in action.
    - The tool should allow users to start, pause, and reset the visualization process.
4. **Performance Analysis:**
    - Measure the execution time of each algorithm with arrays of various sizes and conditions using Python's **time** module.

- Document your findings and prepare a comparative analysis.

5. **Presentation Preparation:**
    - Summarize the project's objective, methodology, key findings, and reflections on the efficiency and practical applications of the sorting algorithms.
    - Prepare a live demo of the visualization tool to include in the presentation.
6. **Demo Video:**
    - Record a video of the live demo showing the sorting algorithms' visualization.
    - Upload the demo video on LinkedIn with a post explaining the project and its significance. Include the hashtag #CSUF to highlight your affiliation and project theme.
7. **Presentation Submission:**
    - Compile your findings, analysis, and demo into a PowerPoint presentation not exceeding 8 minutes.
    - Ensure the presentation is well-structured, engaging, and includes all necessary information to understand your project fully.

PowerPoint Presentation Content

1. **Title Slide:**
    - Project Title
    - Team Members' Names
    - Course Information
2. **Introduction Slide:**
    - Brief overview of the project objective.
    - Importance of sorting algorithms in computer science.
3. **Algorithms Overview Slide:**
    - Brief description of each sorting algorithm implemented (Bubble Sort, Merge Sort, Quick Sort).
    - Theoretical complexity of each algorithm.
4. **Implementation Slide(s):**
    - Challenges faced during implementation and solutions.
    - Key Python libraries used (e.g., **matplotlib**, **Pygame**, **time**).
5. **Visualization Tool Slide:**
    - Overview of the visualization tool and its features.
    - Screenshots or a live demo showing the tool in action.
6. **Performance Analysis Slide(s):**
    - Graphs or tables showing execution times across different array sizes and conditions.

- Analysis of how each algorithm's performance scales with data size and condition.

7. **Conclusion Slide:**
    - Key takeaways from implementing and analyzing the sorting algorithms.
    - Reflections on the efficiency and practical applications of each algorithm.
8. **Q&A Slide:**
    - Invite questions from the audience.
9. **Acknowledgments Slide:**
    - Credits for any external resources or references used.

Tips for Presentation:

- Ensure clarity and conciseness in each slide to keep the audience engaged.
- Use visuals (charts, code snippets, screenshots) to make complex information more accessible.
- Practice the presentation to manage time effectively and anticipate potential questions.

Deliverables

- Source code for sorting algorithms and visualization tool.
- PowerPoint presentation including a live demo.
- A LinkedIn post with the demo video and a brief description of the project, using the hashtag #CSUF.

This project is designed to offer a comprehensive learning experience, from theoretical understanding and implementation of sorting algorithms to the practical application of presenting and sharing your work with a broader audience. It encourages collaboration, critical thinking, and effective communication, skills essential for success in computer science and beyond.

Evaluation Criteria

1. Implementation of Sorting Algorithms (6 Points)

- **Correctness and Efficiency (3 points):** Code accurately implements the sorting algorithms without errors. Algorithms should exhibit the expected time complexity.
- **Code Quality and Documentation (3 points):** Code is well-organized, commented, and follows good programming practices, making it easy to read and understand.

2. Visualization Tool Development (6 Points)

- **Functionality (3 points):** The visualization tool accurately demonstrates the sorting process for each algorithm. Features for starting, pausing, and resetting the visualization work as intended.
- **User Interface and Experience (3 points):** The tool is user-friendly, with an intuitive interface that makes it easy to switch between algorithms and understand what is being visualized.

3. Performance Analysis (6 Points)

- **Comprehensiveness (3 points):** Analysis covers different datasets sizes and conditions, providing a thorough comparison of the algorithms' performance.
- **Accuracy and Insight (3 points):** The analysis accurately reflects the performance of the algorithms, with insightful explanations of the results and their implications for algorithm selection.

4. Presentation and Demo (6 Points)

- **Content and Organization (3 points):** Presentation content is well-organized, covering all project aspects clearly and concisely. The structure follows a logical flow that is easy to follow.
- **Delivery and Demo (3 points):** The presentation is delivered confidently within the time limit. The live demo is effectively integrated into the presentation, showcasing the project's practical outcomes.

5. LinkedIn Video Post (6 Points)

- **Quality and Clarity (3 points):** The video is of high quality, with clear audio and visuals. The demonstration of the visualization tool is clear and effectively showcases the sorting algorithms in action.
- **Engagement and Communication (3 points):** The post clearly explains the project's purpose and findings, engaging the audience with insightful content. Use of the hashtag #CSUF effectively associates the project with the university.

Total Points: 30

Additional Notes for Students:

- Ensure all project components are thoroughly tested before submission.
- Practice your presentation multiple times to ensure it fits within the 8-minute time frame.
- Engage with your audience on LinkedIn by promptly responding to comments and questions about your project.

This evaluation criteria aim to guide students in focusing their efforts on areas that are crucial for demonstrating their understanding and skills in implementing and analyzing sorting algorithms, as well as effectively communicating their findings.

Important Note: *Remember, the journey is as important as the destination. This project offers a chance to not only create a valuable tool but to learn and grow as software developers and team players. Best of luck!*