

✓ COSC2658 - Data Structures and Algorithms/COSC2469 - Algorithms and Analysis/COSC2203 - Algorithms and Analysis

Group Project (REAL)

Assignment Overview:

This assignment challenges you to design and implement an efficient algorithm to crack a 12-character secret key comprised of the characters 'M', 'O', 'C', 'H', and 'A'. The process involves guessing the key and refining the guesses based on feedback from a provided `guess()` function, which returns the count of matched positions with the correct key. The goal is to discover the secret key with the least number of guesses, demonstrating the application of various algorithmic design paradigms and data structures.

Highlights and Advice for the Secret Key Guesser Assignment:

Highlights:

1. **Clever Use of Feedback:** Unlike typical brute force, your algorithm must use the feedback from `guess()` to eliminate impossible combinations, similar to the game of Mastermind.
2. **Algorithmic Paradigms:** This problem allows you to demonstrate understanding and application of various algorithmic design paradigms such as brute force initially and then optimizing using perhaps a decrease and conquer strategy by eliminating known incorrect options.
3. **Custom Data Structures:** You might need to devise a custom data structure to store partial solutions or guesses that could be efficient in updating and iterating through possible solutions.
4. **Performance Measurement:** Since the number of guesses is a performance metric, you'll need to ensure your solution is not just correct but also efficient.
5. **Practical Application:** This simulates a real-world problem where efficiency and smart computing can save significant resources.

What-to-Avoid:

1. **Brute Force Reliance:** Do not rely solely on brute force; use it as a starting point but quickly employ smarter strategies to minimize the guess count.
2. **Ignoring Feedback:** The `guess()` method's feedback is critical for refining your search. Ignoring it will result in inefficiency.
3. **Complex Data Structures:** While custom data structures are encouraged, avoid overly complex structures that could introduce unnecessary overhead.
4. **Lack of Testing:** Do not overlook thorough testing with different cases, as it's crucial for ensuring your program's robustness and efficiency.
5. **Inefficient Loops:** Avoid nested loops or repetitive checks that can greatly increase the number of operations.

Advice for Writing README.txt:

- **Clarity:** Ensure that the README.txt is clear and concise, providing all necessary information without any ambiguity.
- **Structure:** Organize the content in sections, including installation instructions, usage guidelines, testing procedures, and contact information.
- **Contribution Scores:** Clearly define the roles and contributions of each team member if it's a group project.
- **Video Link:** Include a working link to your video demonstration and ensure it adheres to the length and content requirements.

Advice for Video Presentation:

- **Preparation:** Outline your presentation beforehand and ensure it showcases the assignment's objectives and your implementation.
- **Demonstration:** Show a live demo of your program working, including the output of the `guess()` method and the number of guesses taken.
- **Explanation:** Explain the algorithms and data structures you developed, and discuss the rationale behind your approach.
- **Complexity Analysis:** Provide a brief complexity analysis of your solution in terms of both time and space.
- **Results:** Discuss the results of your testing and how it aligns with the assignment's objectives.
- **Editing:** Edit the video to make it as professional as possible, removing any errors or unnecessary parts.

Next Steps:

Based on the assignment details, the next steps would involve creating the `SecretKeyGuesser` class, developing an algorithm that efficiently utilizes the feedback from the `guess()` method, and ensuring that the code adheres to the specified requirements without using the Java Collection Framework. After the implementation, you would move on to preparing the README.txt, technical report, and video demonstration.

> Solution

↳ 2 cells hidden