CMSC389O Midterm Rubric

Expect a thirty minute virtual interview with one of the TAs. You will video chat via Google Hangouts (video not required), while writing code in a shared Google Doc. You will be asked one regular-difficulty coding question. In addition, to warm up, your TA will ask you one short behavioral question at the beginning of the interview. Prior to your midterm interview, please send your resume to your interviewer.

The interview is structured as follows:

- 5-10 minutes behavioral
- 20-25 minutes technical question about one of the following topics
    - Bit Manipulation
    - String Manipulation
    - Arrays
    - Sorting and Searching
    - Linked Lists
- Big-O Analysis of your code, and Test Cases will be required throughout

Behavioral Question (10)					
Behavioral Interview Question (10)	Student answered the question in great detail and was able to tie in many of their skills in the answers. (10)	Student answered the question in great detail and was able to tie in some of their skills in the answers. (8)	Student answered the question. (5)	Student did not fully answer the question. (3)	Student was unable to answer the question. (0)
Technical Question (90)					
Code (50)					
Clarity (5)	Code was extremely easy to read. (5)	Code was easy to read. Very few explanations were needed. (4)	Code was challenging to read. But, Interviewer had complete understanding of what the code was doing, after short explanation. (3)	Code was difficult to read. Interviewer had difficulty understanding what the code was doing, even with in-depth explanation. (2)	Code was impossible to read. Interviewer had no understanding of what code was doing, even with in-depth explanation. (1)
Syntax Mistakes (5)	No syntactic mistakes. (5)		Very few syntactic mistakes. (3)		Many syntactic mistakes. (0)

Logic Mistakes in Code (5)	No logic mistakes. (5)	Some logic mistakes, but noticed immediately when started running through examples. (4)	Some logic mistakes that required prompting to notice but students immediately understood their mistake. (3)	Some logic mistakes that required prompting to notice and students needed in depth explanation to understand their mistake. (2)	Many logic mistakes. (1)
Algorithm Correctness (25)	Overall logic for solving the problem was correct and well thought out. (25)	Overall logic for solving the problem was mostly correct and worked well with minor adjustments. (20)	Overall logic for solving the problem was rocky at first but with promptying from the instructor the student was able to quickly come up with the correct logic. (15)	Overall logic for solving the problem was rocky and even with promptying from the instructor the student had a hard time coming up with the correct logic. (10)	Overall logic for solving the problem was rocky and even with promptying from the instructor the student was unable to come up with the correct logic. (5)
Time/Space Complexity (10)	Same as HW, the most optimal space and time gets you 10 points, and you will lose 2 points per worsening time or space complexities.				
Soft Skills/Other (30)					
Communication (5)	Student spoke clearly and was easily understood all of the time. (5)	Student spoke clearly and was easily understood most of the time. (4)	Student spoke clearly and was easily understood some of the time. (3)	Student did not speak clearly and was not easily understood. (2)	Communication became a barrier to the instructor's understanding of what the student was doing. (1)
Engagement with Interviewer (10)	Student actively engaged with interviewer by asking for understanding and explaining their thought process. Explains all relevant information like edge cases without prompting.(10)	Student actively engaged with interviewer by explaining their thought process. (8)	Student actively engaged with interviewer by explaining their thought process, but needed prompting at times. (6)	Student only engaged with interviewer when prompted, but gave full and complete answers. (4)	Student only engaged with interviewer when prompted and gave very brief responses. (2)

Test Cases (5)	Student runs through all relevant test cases. (5)	Student runs through most relevant test cases. (5)	Student runs through some relevant test cases. (5)	Student runs through one relevant test cases. (5)	Student runs through no relevant test cases. (5)
Edge Cases (10)	Student addresses all relevant edge cases and their code handles it. (5)	Student addresses all relevant edge cases, but their code does not handle all of it. (4)	Student addresses most relevant edge cases and their code handles it. (3)	Student addresses most relevant edge cases, but their code does not handle all of it. (2)	Student does not address any edge cases. (1)

Tips To Rocking Each Section:

- Behavioral Interview Question
    - Take your time answering the question.
    - Make sure you answer all parts, have a clear narrative, and finish your answer.
    - If the interviewer is quiet on the other end, stay confident in your response. They are probably writing down notes.
    - Think about what questions your interviewer may ask beforehand, and prepare some responses. These can generally be applied to many different types of questions.
    - Know your resume inside and out! Be prepared to talk about a project or internship that you have listed on your resume. Think about the exercises we did in the first week!
- Technical Interview
    - Code
        - Clarity
            - Spacing out code well
            - Following standard coding practices for your language
            - Clear comments where needed
        - Syntax Errors
        - Logic Errors in Code
            - Think about edge cases while writing code
        - Algorithm Correctness
            - Make sure your solution addresses the problem
- Soft skills
    - Communication
        - Speaking clearly means talking with an even tone, not too fast, not too quietly, and with word choice that has your audience in mind.
    - Engagement with the Interviewer
        - Things you should be doing
            - Explaining your thought process outloud to interviewer.
            - Checking that the interviewer understands what you are doing.
                - Ask questions like, do you have any questions? Do you understand what I mean here? Etc.
            - Take the lead in the conversation (e.g. when you are done coding, tell your interviewer that you would like to run through some test cases; talk about time and space complexity without prompting)