Self-Assessment Scorecard

Flip over for details

[0:00 - 0:05 mins] Understanding the problem

□ Arrived at example, inputs and outputs. Described what you see vocally
□ Clarified input & output formats
□ Clarified return errors, exceptions, if any
□ Outlined edge cases, if any

[0:05 - 0:10 mins] Brute Force approach.

□ Arrived at a brute force approach
□ Described complexity of the brute force approach
□ Declared that you will see how to optimize this further

[0:10 - 0:25 mins] Optimal solution, pseudocode and complexity

□ Arrived at an optimal solution with pseudocode
□ Pseudocode walkthrough with at least 2 of the examples from step 1
□ Communicated optimal space-time complexity to the interviewer

[0:25 - 0:35 mins] Code

□ Wrote the code for the optimal solution
□ Check code for any syntax & logic errors and typos. There should be very few if any
□ Write down the optimal space-time complexity on the whiteboard

[0:35 - 0:40 mins] Walkthrough

□ Code walkthrough with atleast 2 examples
□ (Optional) Brief note of testing (Spend between 30s to 60s here and just verbalize the positive, negative and edge tests you can perform)

[0:40 - 0:45 mins] Buffer & questions for the interviewer

□ Asked questions to the interviewer about the company & job
□ Thanked them for the interview

How to use this Self-Assessment Scorecard

Flip over for the scorecard

This scorecard is used to assess yourself after an interview. We've described the minimal set of checkpoints and timeline of a 45 minute interview. Successful interviews hit most of these checkpoints and do so at the right time intervals. In addition to these checkpoints the below should be true for all 45 minutes of the interview:

1. Believe that it is your responsibility that the interviewer knows what you are thinking and doing. The onus is on you to ensure they are not left behind
2. Lead the interview as much as possible, do not wait for the interviewer's approval for any step. There are good questions but these are quite limited to the problem clarification phase.
3. Stay in control of the time, whiteboard and your pace.

Use this scorecard in mock interviews, real interviews and practice with it at home. The goal is to gain control of the time that flows from start to finish of the interview. This will let you stay calm and realize that you have a plan. If you are falling behind on the timeline, always opt to salvage what is possible and reach the end. Carry a digital sports watch.

[0:00 - 0:05 mins] Understanding the problem
In this stage you haven't begun solving the problem yet. Assuming the solution is a black-box you arrive at sufficient examples of the possible inputs and outputs to the solution and you clarify the possible formats for inputs and outputs, errors, exceptions and edge cases. If the interviewer gives you more than 2 examples of inputs & outputs you should use this time to examine these examples instead of coming up with your own.

[0:05 - 0:10 mins] Brute Force approach.
In this stage you arrive at the bare minimum, brute force approach and describe its complexity. This shouldn't take much time. You've spent some time already looking at various inputs and outputs and at this stage you must describe a bad solution to the problem

[0:10 - 0:25 mins] Optimal solution, pseudocode and complexity
The bulk of the interview should be spent in this stage. This is your problem solving stage. At this stage you want to make *small, good, incremental decisions* to arrive at core insights that will lead you to the solution. The more you describe your mental thoughts and thinking process in this phase, the more likely you'll get hints from the interviewer. Resist any temptation to jump into coding. Use the following signals to determine when it is time to start coding:

1. Describe the complexity of an optimal solution to the interviewer and ask them if it is acceptable. Their response is a signal to continue optimizing or declare that you'll start coding the solution
2. You have sufficient amount of pseudocode on the whiteboard and you can visualize what the entire code is going to look like

[0:25 - 0:35 mins] Code
Coding should be the least of your worries. When you get to this stage and if you did with the signal from the interviewer that your complexity was good, then this is your finishing lap. You need to execute well but your preparation should have ensured that your coding skills are the least of your worries. In addition, your pseudocode from the previous section has solved all of the logical complexity needed and all you need to do is translate that pseudocode into real code. It should feel like you banged out the function top to bottom in a few

minutes and spent some time looking for typos, syntax and basic logic errors. End with writing the complexity of the optimal solution under your code and state that this was discussed previously.

[0:35 - 0:40 mins] Walkthrough

In this section, choose 2 examples and perform a walkthrough of your code on the whiteboard. If you have 30-60 seconds talk about testing. Briefly touch upon positive, negative and edge tests that would be valuable to perform at this stage

[0:40 - 0:45 mins] Buffer & questions for the interviewer

This is your buffer time. If you get here completing all the above, then you can use this time to express enthusiasm by asking interesting questions, and generally end the interview on a happy positive note.