Vasudev Agarwal
vasu14devagarwal@gmail.com

Technical Interview Week: Feedback Rubric [create a copy]

CODE PATH *ORG

Feel free to share a completed copy of this rubric with your student by email after the interview!

UMPIRE Method from CodePath is the solution for the critics listed above

Question	3Sum
Category	TIP102
Problem Solving • Did the student ask clarifying questions to gather more information before trying to solve the problem? • Was the student able to come up with multiple solutions and explain their trade-offs?	Vasudev did a good job of asking clarifying questions and was able to come to explore alternate solutions and their trade-offs. U: Understand M: Match
Communication • Was the student able to explain their approach in an understandable way? • Was the student able to ask relevant questions to understand the problem? • Did the student talk about their general approach before coding it up? • If the student got stuck at any point, were they able to talk through why they got stuck?	Vasudev was able to articulate their general approach in an understandable way and stated when they were stuck. They asked relevant questions, but should make sure to say what they're thinking when encountering problems. For example, it was clear from early on that Vasudev had figured out their conceptual approach, but there was a missed opportunity to ask an important clarifying question when they were not sure about dealing with the order of the input list. I offered a hint about sorting after Vasudev stated they were stuck on figuring out how the pointers should traverse the list. P: Plan/ Pseduo Code I: Implement
Coding Skills • Was the student able to code up a working solution? • Did the student write easy to understand code? • Did the student consider possible edge cases and handle it in their code? • Was the student able to verify their code works by tracing through the code by hand or by running tests?	We did run out of time, but Vasudev had what looked like working code by the end. We identified a couple areas of improvement (utilizing a set to handle the exclusion of duplicate triplets, remembering to return the list). The code was easy to follow. There were some naming issues ('left'/'right' became 'l'/'r'), but they didn't hinder my ability to understand what they wrote. We discussed edge cases, but this was after I asked them to identify some. Their kSum approach sounded very cool, and it was obvious that they were excited by the problem. I would have enjoyed seeing them work through that problem as well. We were unable to test the code, but they did run through it by hand. P: Plan/ Pseduo Code I: Implement R: Review (for edge cases)

• If there were bugs, was the student able to notice and fix the bugs themselves?	
Evaluation Skills • Was the student able to correctly state the run time and space time complexity of their solution?	Yes. P: Plan/Pseduo Code
Speed • Was the student able to complete the question within the time frame? • Did the student spend too much or too little time asking questions before trying to code up a solution?	We ran right up to the time limit. I think they could move more quickly into writing the code. We spent a good chunk of time talking through the problem and writing pseudo-code, and I think we would have benefited from more time coding. Sometimes getting into writing the code will help us uncover more clarifying questions. Took too much time in 'P', limit to 8 - 10 mins for UMP
Response to feedback • If you gave the student any feedback or hints, was the student able to take the feedback and improve their solution?	Vasudev was very receptive to feedback and hints. They immediately recognized the value in using a set to handle the duplicate triplets.
Final Evaluation:	Overall, Vasudev did a great job. They correctly identified the solution early on. They asked good clarifying questions and were receptive to feedback. I would recommend making sure to verbalize any problems they're encountering so that the interviewer is able to offer hints, and keep an eye on the clock when discussing the problem and writing pseudo-code so that we don't hurt ourselves by cutting into our actual coding time.

Communicate Issues what you are facing, so that the interviewer can help you

CPU and ABS
CPU: psudeo code should be used as cache memory when you get struck
ABS: Always Be Speaking, what issues you are currently facing