

5. The New Hire...

I decided to take a pragmatic approach to this in the sense that I wanted to get inside the head of each of the candidates and understand their logic.

The first step in achieving this was to understand the problem itself first. Given that I am conducting the hiring, it is inherently assumed that I understand the problem and have (if not come up with a working solution for all levels) made a significant effort to solve the problem myself. This gave me a sense of the level of difficulty involved in what each candidate was endeavoring to do, the amount of time involved in procuring a solution, and the myriad of permutations of “could-be” outcomes.

My next inclination was to get to an idea of each person’s character based on their confidence level outlined in their self-descriptions and their respective approaches to the solution. More points for those whom gave a detailed synopsis vs those with one line of detail.

I proceeded to take each of their solutions and copy/pasted them into the interpreter, running them all to validate the complexity level achieved.

Having established that all solutions work in some capacity, I reviewed each submission taking into consideration variables such as – how well commented is the code? Is it easy to read? “Sometimes more is less”, have they refactored if necessary?

Having given tremendous consideration to all 4, I politely wrote the 3 disqualified candidates and informed them that we wish to move on – awarding Swiss Chris the position.

He noted some of the limitations of the API itself, which in turn, lent to his reasoning for his approach and deprecating some of the API functionality. I awarded him extra points for testing the API itself, as his logic was improved beyond the other candidates as a result of the extra investigative work.

He completed a solution which conforms to the 18th (final) level of complexity.

His solution (code) is very well formed, dividing all portions of the code into regions, for easy readability and less time involved in parsing the code for a given piece of functionality.

His solution is extremely well documented. He has comments for virtually every block of code in the solution.

He is modest/humble. He passed out kudos to other developers who had procured a solution of which he deemed better than his own.