1. Strategies for a great interview

Approaching the problem:

1. **Clarify the question:**
   * What is a concrete example of the problem?
     + Ex: If the question is “find the first occurrence of a number greater than k in a sorted array”, you could ask “if the input array is <2, 20, 30> and k is 3, then are you supposed to return 1, the index of 20?
   * What time and/or space complexity should the solution be?
     + Ex: You are told to implement an O(n) algorithm or use O (1) space
2. **Work on concrete examples:**
   * Ex: Determine the smallest amount of change that you cannot make with a given set of coins. This problem may seem difficult at first.
     + However, if you try out the smallest amount that cannot be made with some small examples, e.g. {1, 2}, {1, 3}, {1, 2, 4}, {1, 2, 5}, you will get the following key insights:
     + Examine coins in sorted order, and look for a large “jump” a coin that is larger than the sum of the preceding coins
3. **Spell out the brute-force solution:**
   * Something is better than nothing
   * Tech interviews problems tend to have an obvious brute-force solution with a high time complexity compared to more sophisticated solutions
   * Why Brute-Force?
     + Helps you explore opportunities for optimization and hence reach a better solution
     + It gives you an opportunity to demonstrate some problem solving and coding skills
     + It establishes that both you and the interviewer are thinking about the same problem