A THINKING APE

VANCOUVER

Congratulations once again, we're super excited to be moving forward with your application! The following document includes some tips that we've put together to better prepare you for your upcoming interview.

What to Study for a Technical Interview

Keep in mind that the interviewers are looking at how you approach and solve the problem; it's as much about your work as the final solution. We encourage you to think out loud and ask questions. Make sure your interviewer knows where you are at! (i.e: if you require a hint, if you're struggling with a specific part of the question…). Our questions are designed to expand in many different directions! We want to see how you think when you are challenged and pushed.

Below are some of the topics that will be beneficial for you to study up on before your technical interview:

Algorithms

- It is always beneficial to understand many modern, well known algorithms, as variations of them can often help solve a complex problem.
- Make sure you are confident with recursion.

Data Structures

- Most candidates benefit from knowing hashmaps, queues, stacks, lists, trees (Binary Search Tree, Min Heap, etc.).
- Additionally, being confident in the runtime of these data structures is important. Which ones provide fast lookups, insertions, removals, etc.
- Finally, many questions are best answered using a combination of these structures. Thinking of ways to add to known implementations of the structures to speed up operations, or ways to use more than one in combination can be very beneficial.

Programming

- Often design patterns can be extremely useful. (Observer Pattern, Singletons, Factories, etc.)
- Make sure you are confident on how to define and instantiate objects, and how to create global functions and variables. (While this is easy enough to look up, intrinsic knowledge often shows confidence in your chosen language.)

Interviewing

- It is ideal to describe the meat of your solution with full syntax, including every expression, block and bracket. It's okay to gloss over boilerplate code, especially if you have written it once already in an interview. (Say write 1 full object construction, and use pseudo-code for the rest to save time). we're not testing for an encyclopedic knowledge of any given language. If you have to make any assumptions about API calls or library functions (e.g. 'Is it add or append for lists in python?'), just state your assumption. Anything you can google is fair game to ask us.

Steps To Go Through In An Interview

- Your interviewer will present the problem to you. Ask questions, clarify, make sure you fully understand the problem before diving in.

- Talk through the first solution that comes to mind, ask if you should implement it
    - Usually brute force, feel free to say you will refine it
    - Run through examples to check - assume someone else wrote the code
    - Check for edge cases
    - We can't evaluate your code if you don't code!
    - Give your interviewer the opportunity to ask question about your solution

- Refine the solution
    - Clarify assumptions
    - Interviewer will likely have questions for you at this point
    - Can you improve the run time?

PRACTICE

- "Cracking the Coding Interview" book
- HackerRank, LeetCode, CodeLab, Quora, Stack Overflow, Interviewing.io
- Review algorithms and data structures - know how they work and why you choose one over the other:
    - Quicksort, merge sort
    - A*
    - Hashtables
    - Queues
    - Stacks
    - Linked lists
    - Binary search
    - Trees (binary search tree, min heap, etc.)
    - Breadth-first search, depth-first search
    - 2D arrays
    - Dynamic array
    - Dynamic programming
    - Big-O analysis

- This link is really good https://www.topcoder.com/community/data-science/data-science-tutorials/. In particular: Greedy is Good, Data Structures, An Introduction to Recursion, and Dynamic Programming.

- Practice writing code! Pick your strongest language and practice that
    - Use a whiteboard or a piece of paper

Remember to talk through solutions and ideas, ask questions, your interviewer wants to help you! If you're unsure about something, just ask!!