OCI Technical Interview Preparation

No matter your previous experiences at other employers, at Oracle as part of team OCI, you are here to build something great and not be managed by checklists. We thought the resources listed below may help before your interview here. Even for someone with lots of experience, who has not interviewed in a long time, these "refresher" suggestions can help you get into the zone and help facilitate communication during your interview day. It's not expected that you study for hours before an interview with us. But you should expect coding questions, typically asked and answered on a whiteboard.

Topics

Here are preparation topics:

- Data structures: lists, stacks, queues, trees, hash tables, sets, tries, graphs
- Algorithms: Sorting, Recursion, Searching, Graph and Tree algorithms, Dynamic Programming
- Time and Space efficiency (big-O analysis)
- Problem Solving Practice (try a few exercises for these areas from the books listed below).

Books

If you have it, but haven't opened it in a while, why not scan through it? Should be available at your local library. Pick a few exercises.

- Cracking the Coding Interview
- Introduction to Algorithms – CLRS is the classic, but can be dense. Consider augmenting with one or more of:
  - The Algorithm Design Manual - Steven S. Skiena
  - Algorithms (4th ed.) – Sedgewick and Wayne
  - Algorithms – Dasgupta et al.
- Programming Pearls - Jon Bentley
- open data structures textbook: http://opendatastructures.org
- Jeff Erickson's algorithms notes (book-length): http://jeffe.cs.illinois.edu/teaching/algorithms/
- The Art of Computer Programming - Donald Knuth (way overkill for interviews)

Other Technical Background:

Including distributed systems topics:

- Intro level videos: Cloud Computing Concepts. Sections 4 and 5 are most relevant.
- REST API best practices (various sources on the web).
- Paxos algorithm: https://www.youtube.com/watch?v=WX4gjowx45E
- CAP theorem and NOSQL: https://www.youtube.com/watch?v=LW8MBYU_pzQ
- Martin Fowler's talk on Intro to NoSQL: https://www.youtube.com/watch?v=qI_g07C_Q5I
- MIT OCW Introduction to Algorithms: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011 (the video lectures are excellent)

Other Good Resources:

- Glassdoor.com - Common interview questions asked by companies
- Careercup.com - Interview questions
- Behavioral Questions - https://www.flazingo.com/blog/2014/02/21/behavioral-interview-questions/

- Steve Yegge's blog on interviewing: http://steve-yegge.blogspot.com/2008/03/get-that-job-at-google.html
- Practice problem sites: https://hackerrank.com, http://www.spoj.com, https://uva.onlinejudge.org

Practical Tips

- Ask clarifying questions about requirements. Sometimes there are ways to dramatically simplify a problem if you can impose a restriction on input, for example.
- Try to talk through your plan before jumping into code. Remember, we're mainly trying to figure out how you think about solving problems, not just how you write code.
- Going with the simple, brute-force, sub-optimal solution and then iterating on it for time and/or space efficiency is perfectly acceptable. In the worst case, having something working that you can give reasonable suggestions for improving beats having an incomplete implementation of the
perfect solution.
- Once you've written out something, take a moment to come up with a simple test case and walk through it, time permitting. Even if you don't have time to fully walk through a test run, take a minute to talk about how you would test your code.
- When you're practicing, try writing out solutions long-hand on paper. This better mimics the conditions of an interview in most cases than using an IDE with code-completion. Things you have to struggle to remember are good things to review.
- If the list of material presented above is intimidating in scope or something you are not reasonably fresh on, it's ok to take time to brush up before continuing the interview process.

Good Luck!