TECHNICAL INTERVIEW - WORKSHOP

SANDHIYA RAMESH

MASTERS IN INFORMATION SYSTEMS

ROLES AND DESCRIPTIONS

- Software Engineer
- Software Developer
- Full Stack Developer
- Data Analyst / Business Analyst
- Data Scientist

PROGRAMMING INTERVIEW

SOFTWARE ENGINEER / DEVELOPER INTERN

- Coding
- Algorithms
- Data Structures

TYPES

- Questions involving loops, lists, or strings (easier)
  - *For example, Remove duplicates from a list or Generate all of the permutations of a string.*
- Questions involving specific data structures (harder)
  - *For example, if you've never done breadth-first tree traversal, and you're asked: Print the values in this tree level-by-level. Note: Big-O complexity of your implementation***QUESTIONS**

"WHAT HAVE YOU DONE?"

- *What was the project?*
- *What was the scope of the project? (i.e., how much time, how big was it)*
- *What was your role in the project?*
- *What did you learn?*
- *What were some of the challenges in the project?*

MOST COMMON QUESTION

Tell me about a technical challenge that you had.

What makes {language} different from other languages?

If you had to change something in {language}, what would you change?

Tell me about a technical challenge that you had when you were working on a project in {language}

BRUSH UP TOPICS

DATA STRUCTURES

- *Linked Lists*
- *Trees*
- *Hash Tables*
- *Stacks*
- *Queues*
- *Graphs*
- *Tries*

ALGORITHMS

- *Breadth-first search (DFS)*
- *Depth-first search (BFS)*
- *Mergesort*
- *Quicksort*
- *Binary Search*

DATA STRUCTURE Operations

www.bigocheatsheet.com

ARRAY SORTING Algorithms

DATA Structure	TIME Complexity: Average				TIME Complexity: Worst				SPACE Complexity: Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

ARRAY Algorithms	TIME Complexity: Best	TIME Complexity: Average	TIME Complexity: Worst	SPACE Complexity: Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(1)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)t	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

Elements **O(1), O(log n)**

COMMON INTERVIEW QUESTIONS

JAVA

- Java Internal Working
- Internal Working of Hashmap and HashSet
- How to compare two objects in Java?
- How to sort user objects in Java?
- Comparator vs Comparable?
- Serialization and Deserialization?
- Immutable class?
- Deep vs shallow Copying
- Reflection in Java

- **FRAMEWORK:** Spring MVC, Spring Boot, Hibernate, REST API

MULTITHREADED PROGRAMMING

Producer and Consumer
Blocked Queue
Countdown Latch
Cyclic Barrier
Runnable thread
Thread pool
Callable and Future
Deadlock
Immutable Class
Volatile
Reentrant lock
Semaphore
Deadlock avoidance

COMMON INTERVIEW QUESTIONS

SOFTWARE DEVELOPMENT TECHNOLOGIES

- Agile Scrum methodologies - JIRA
- Java Software Design Patterns(Singleton, DAO, Factory, Adaptor, Spring MVC)
- How Waterfall Model different from Agile Methodologies?

CLOUD COMPUTING – AWS

- ELB
- EC2
- Cloud Watch
- IAM
- Linux commands

Example of an application hosted in cloud and challenges you faced

AREAS TO FOCUS PER ROLES

Front End Developer

- Javascript: Closure, Promise, Callback Functions, Difference between === vs ==, Internal working of Javascript
- Why and where used AJAX?
- Framework: MEAN(Angular4), ReactJS
- CSS3, HTML5
- MongoDB(NoSQL), MSQL

Note: Project Explanation and push in GitHub

COMMON INTERVIEW QUESTIONS

DATABASE

Normalization (3NF)

What do you mean by data manipulation language?

What do you mean by data definition language?

What do you mean by Transaction Language?

What are the differences between DDL, DML and DCL in SQL?

What is the difference between primary key and unique constraints?

What are super, primary, candidate and foreign keys?

SQL BASICS

What is a view in SQL?

Functions vs Stored Procedures?

Triggers?

What is an index? or How will improve the performance of the database?

What are clustered and non-clustered Indexes?

What is the difference between having and where clause?

Why we cannot use WHERE clause with aggregate functions like HAVING ?

What is the difference between BETWEEN and IN operators in SQL?

What is a join in SQL? What are the types of joins?

What do you mean by foreign key?

What is the difference between CHAR and VARCHAR?

What is a transaction? What are ACID properties?

REFERENCES

Books to Refer: Links to Refer:

Cracking the Coding Interview: 150 Programming Questions and Solutions

Programming Interviews Exposed

Links to Refer:

https://www.javatpoint.com/collections-in-java
https://www.programcreek.com/2012/11/top-10-algorithms-for-coding-interview/
https://www.programcreek.com/wp-content/uploads/2012/11/coding-interview2.pdf

Adobe Acrobat Document

Microsoft Word Document