MinT Technical Interview Examples

About this Document

This document provides sample technical interview questions and solutions for mentors to use when conducting Mock Technical Interviews with their mentees. This document is a companion to the Mock Technical Interviews outline.

This document is not meant to be shared with mentees. Remember that your mentees will also be completing mock interviews with their other mentors.

The questions in this document are provided as *examples*. You are encouraged to use questions from your own experience instead or as well.

Tips for Interviewers

Selecting Questions for mock interviews

As designed, students in the MinT program will experience 2 mock interviews with each of their 2 mentors, for a total of 4 interviews. We recommend that you start your first interview with a straightforward "screening" type question. If they can handle that question well, move on to more complex open-ended questions.

We've included a few questions on architecture and on user experience design. The architecture questions should be considered as very advanced. The UX/design questions should be given only to students who have explicitly expressed an interest in these types of roles.

This document is just a starting point and reference to calibrate difficulty. You are encouraged to use questions from your own experience as an interviewer or interviewee.

General Follow-up Questions

This incomplete list of follow-up questions should be generally applicable to any programming-oriented question:

1. Why would you use X data structure to solve the problem? What are the Pros/Cons of that DS?
    - Interviewers are generally looking for the candidate's ability to explain the tradeoffs between technologies and why one data structure or algorithm is better for this particular problem (usually because of space or time complexity), as well as their knowledge of specific data structures.

2. How would you test this code? How might you make your code easier to test?
    - Looking for the candidate's ability to run through their own code with an example and catch their own bugs
    - Also looking for them to build modular, maintainable, easy-to-test interfaces: i.e. don't put all your logic in one method.

3. What is the runtime of your code? How might you improve your runtime?

- Looking to understand the candidate's ability to know their code and know runtimes of standards data structures and algorithms if they are utilizing them.

4. What is the space complexity? How might you improve your space complexity?
    - Less important, but usually called out as a tradeoff if you are trying to improve your runtime.

5. Please test this code
    - Ideally, the candidate does this without prompting, which would indicate that they are in the habit of validating their code
    - Looking for understanding of the requirements
    - Looking to see how they approach debugging

Screening-type Questions

1. Write a method that reverses a string delimited by spaces.

    Example
    ```
    Input: my name is kevin
    Output: kevin is name my
    ```

2. Given two arrays of chars, A and B, return an array that consists of the chars in array A excluding all occurrences of any char that appears in array B.

    Example
    ```
    A: my name is kevin
    B: ne
    Output: my am is kvi
    ```

3. What's your favorite data structure? Why do you like it? How does it compare to *[other data structure]*?

4. When should you use a hash-table vs binary search tree? What if storage is not a concern? What if computing operations were fast?

5. Model a binary tree that uses these numbers: 2, 7, 5, 6, 9, 2, 4, 11, 5. Is your tree a Binary Search Tree? Why or why not?

6. Write a program to reverse a linked list
    Solutions available: [How to Reverse a Linked List | Baeldung on Computer Science]

7. Given a list of numbers, find n numbers in the list that sum to value y

    Example

```
list: [42, 8, 1, 3, 1, 2, 5, 7, 11, 4, 15]
n = 3
y = 15
Possible Result: 8, 3, 4
```

Open-ended Questions

Programming-oriented

1. Given an NxM grid of islands where each section is marked with either a ‘L’ or ‘W’ representing Land or Water and all orthogonal-touching ‘L’s are one island, write a method that will return the total count of islands.

2. Write a tool to serialize and deserialize a general tree (each node can have different numbers of children)

3. Discuss ways to implement a queue and implement it

4. Discuss ways to implement an LRU cache and implement it

Architecture-oriented

1. Design the backend/API for a URL shortening service
   solution available: https://www.educative.io/courses/grokking-the-system-design-interview/m2ygV4E81AR

2. Design a system that can play card games.
   a. "Does it need to support Magic The Gathering?" "No, just things you can play with a regular 52 card deck and jokers."
   b. Goal is to get an understandable API with reasonable objects.

Design / UX / Product oriented

1. Design the system that would control the street lights at a standard 4-way intersection
   a. How do the lights stay synchronized?
   b. What external data sources would you use? (Sensors, traffic patterns, time of day, etc.)
   c. How would you detect failure?
   d. How would you scale the system to other intersections that may not follow the same layout?

2. Design a 21st-century vending machine
   a. what kinds of things can it sell?
   b. What payment methods should it accept?
   c. Can it be intelligent about maintaining stock of products and reporting analytics?
   d. Who is the customer and what’s the business model?

3. Design an algorithm for routing elevators
   a. How many elevators?
   b. What to optimize for?

c. How to manage tradeoffs between efficiency and user-control?