Back to Explore (/explore/)

Favorite

Get Well Prepared for

Google Interview

Overview

Google tech interviews are notoriously difficult and quite challenging. To get a phone screen, you will need to submit your resume to their online application system or via an internal referral from a Googler. Assuming you passed their resume screen, a recruiter will reach out to

---

Interview Process

You may receive an online assessment link as your first step of interview process. The assessment will expire within 7 days and contains two coding questions to be completed within an hour. Below are some Online Assessment questions for you to practice. Near the end of this

Arrays and Strings

String manipulation problems are in the same category as arrays, because internally, a string is represented as an array of characters. Array problems usually do not require knowledge of advanced data structures, so just basic data structures such as Hash Tables and basic

Linked Lists

According to our user survey data, Linked List problems are not asked frequently at Google. Perhaps, most linked list problems are not that complex and it is harder to ask follow up and complexity analysis questions Nonetheless, we strongly recommend you to still practice classic

Trees and Graphs

Tree is just a special case of graph. To understand the difference between trees and graphs, you can work on Graph Valid Tree. Graphs are generally breath-first search or depth-first search. The same applies to Trees, but trees never contain cycles. Graphs are generally more

Recursion

Recursion usually involves some kind of backtracking to enumerate all possibilities. Note that Recursion is a more general purpose algorithm. Depth-First search is a specific form of backtracking related to searching tree data structures. Therefore we categorize those

Sorting and Searching

Interval related problems are quite often asked at Google interviews. Similar to "Arrays and Strings", interval related problems can be asked in the context of data stream.

Dynamic Programming

It can be tricky to identify the subproblems and connect them, which is essential in solving Dynamic Programming problems. Dynamic programming is not that scary as you might think, and you can improve your dynamic programming skills by practicing a lot of these

Design

Google loves to ask lots of question variations based on the Iterator pattern, so make sure you are familiar with the concept of iterators and how iterators work in principle. A good way to learn is to read the open source code and try to code it yourself. For example, here is

Others

Here are other type of problems you may encounter in a Google interview, such as Bit Manipulation.

Discuss (/discuss/explore/google)

26 topics - share ideas and ask questions about this card

Introduction

Google tech interviews are notoriously difficult and quite challenging. To get a phone screen, you will need to submit your resume to their online application system or via an internal referral from a Googler.

Assuming you passed their resume screen, a recruiter will reach out to you. Usually there will be two phone screens, and if you do well, you'll be invited to onsite interviews.

Since Google operates at a large scale, be prepared to answer lots of follow up questions on how to scale the algorithm you wrote for multiple machines. Some examples are: Number of Islands (https://leetcode.com/problems/number-of-islands) and Intersection of Two Arrays II (https://leetcode.com/problems/intersection-of-two-arrays-ii/description/).

Interview Process

- [x] Unique Email Addresses
- [ ] Odd Even Jump
- [ ] License Key Formatting
- [ ] Fruit Into Baskets
- [ ] Google Phone Interview
- [ ] Google Onsite Interview
- [ ] Google Hiring Committee
- [ ] Google Offer Review

Arrays and Strings

- [x] Longest Substring Without Repe…
- [ ] Container With Most Water
- [ ] 3Sum
- [ ] Next Permutation

- [ ] Multiply Strings
- [ ] Rotate Image
- [ ] Jump Game
- [ ] Plus One
- [ ] Minimum Window Substring
- [ ] Read N Characters Given Read4…
- [ ] Longest Substring with At Most T…
- [ ] Missing Ranges
- [ ] Next Closest Time
- [ ] Expressive Words
- [ ] Find And Replace in String
- [ ] Maximize Distance to Closest Pe…
- [ ] Valid Parentheses
- [ ] Merge k Sorted Lists
- [ ] Trapping Rain Water
- [ ] Kth Largest Element in an Array
- [ ] Meeting Rooms II
- [ ] Backspace String Compare
- [ ] Minimum Cost to Hire K Workers
- [ ] K Closest Points to Origin

Linked Lists

- [ ] Add Two Numbers
- [ ] Remove Nth Node From End of …
- [ ] Merge Two Sorted Lists
- [ ] Copy List with Random Pointer

Trees and Graphs

- [ ] Binary Tree Maximum Path Sum
- [x] Word Ladder
- [ ] Number of Islands
- [ ] Course Schedule II
- [ ] Count Complete Tree Nodes
- [ ] Longest Increasing Path in a Matrix
- [ ] Decode String
- [ ] Evaluate Division
- [ ] Diameter of Binary Tree
- [ ] Cracking the Safe
- [ ] Robot Room Cleaner
- [ ] Most Stones Removed with Sam…
- [ ] Flip Equivalent Binary Trees

Recursion

- [ ] Word Squares
- [ ] Strobogrammatic Number II
- [ ] Word Search II
- [ ] Android Unlock Patterns
- [ ] Letter Combinations of a Phone …
- [ ] Generate Parentheses

Sorting and Searching

- [ ] Median of Two Sorted Arrays
- [ ] Find First and Last Position of El…
- [ ] Merge Intervals
- [ ] Insert Interval

- [ ] Valid Anagram
- [ ] Count of Smaller Numbers After …
- [ ] Peak Index in a Mountain Array

Dynamic Programming

- [ ] Longest Palindromic Substring
- [ ] Maximum Subarray
- [ ] Best Time to Buy and Sell Stock
- [ ] Maximum Product Subarray
- [ ] Coin Change
- [ ] Split Array Largest Sum

Design

- [ ] LRU Cache
- [ ] Min Stack
- [ ] Serialize and Deserialize Binary …
- [ ] Logger Rate Limiter
- [ ] Insert Delete GetRandom O(1)
- [ ] Design Search Autocomplete Sy…

Others

- [ ] Reverse Integer
- [ ] Candy
- [ ] Isomorphic Strings
- [ ] Strobogrammatic Number
- [ ] Bulls and Cows
- [ ] Range Sum Query 2D - Mutable
- [ ] My Calendar II

- [ ] Jewels and Stones
- [ ] Swap Adjacent in LR String
- [ ] Guess the Word
- [ ] Minimum Area Rectangle