Technical Interview Tips

CS W169A Summer 2020

Interview Timeline

Initial Phone Screen

Coding Challenge

Phone Interview(s)

Onsite Interview

Offer

Sample Leetcode Outline

D.S. - Arrays	#31 Next Perm., #40 Rotate Image, #189 Rotate Array
D.S. - Trees	#101 Symmetric Tree, #114 Flatten BT, #235 LCA
D.S. - Stacks/Queues	#20 Valid Parentheses, #102 Level Order, #232 Implement
D.S. - Disjoint Sets	#547 Friend Circles, (HR) Components in a graph
Algo - Strings	#14 Common Prefix, #49 Group Anagram, #125 Palindrome
Algo - Recursion	#66 Plus One, #98 Validate BST, #257 Binary Tree Paths
Algo - Search	#240 Search 2D Matrix, #695 Island Max Area
Algo - Backtracking	#22 Parentheses, #39 Combination Sum, #46 Permutations
Algo - Dynamic	#62 Unique Paths, #64 Minimum Path Sum

Additional Resources

LeetCode

Hackerrank

CareerCup

InterviewCake

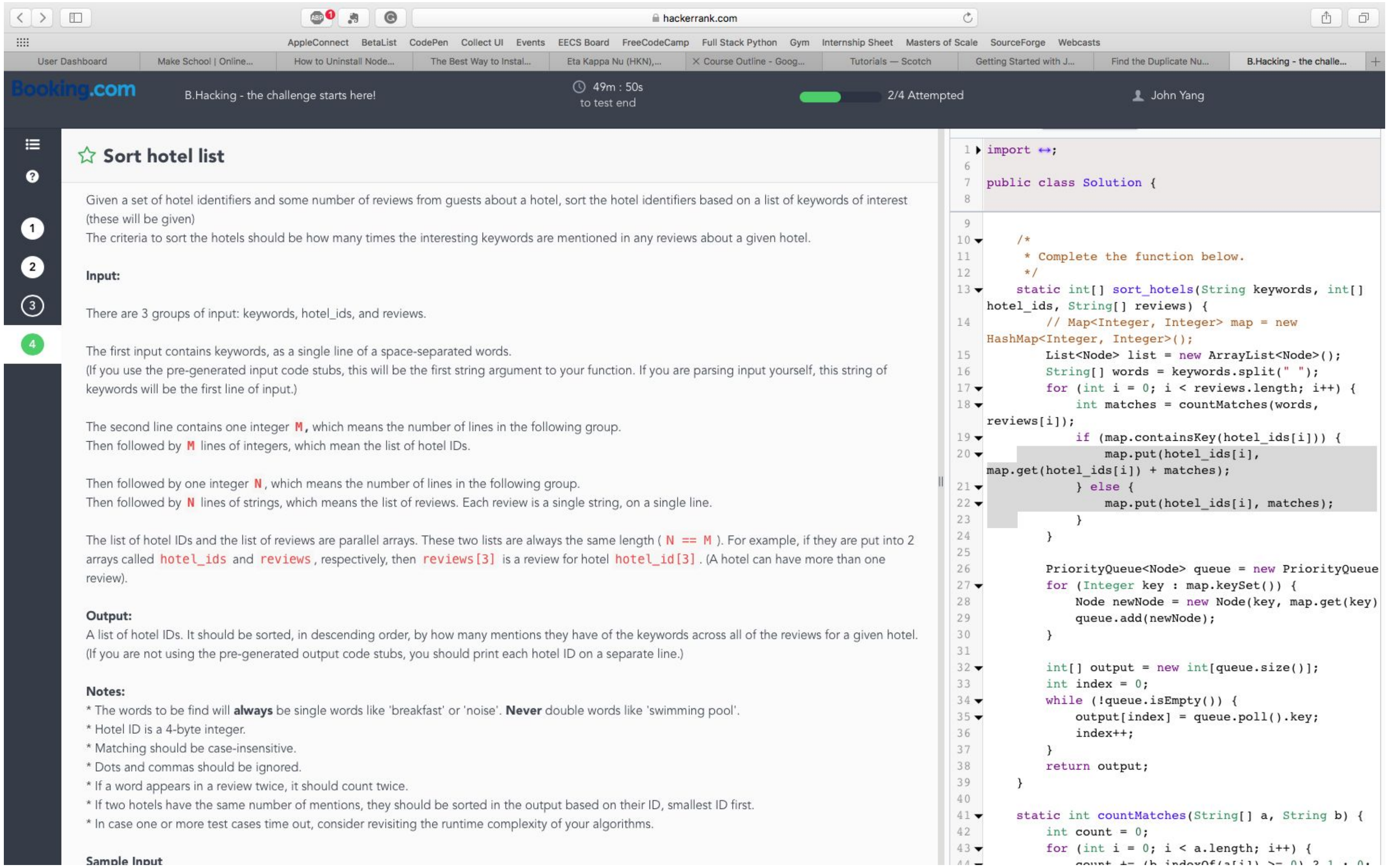
Cracking the Coding Interview

Practice, Practice, Practice!

Coding Challenge Tips

- You will only pass this round if your code passes every available test case AND you complete all problems.
- Typically issued through **Hackerrank**.
- Lasts anywhere between **1** to **3** hours.
- Usually, you can choose your preferred language of choice i.e. Java or Python
- Use **Google!!!** This is not a homework or project. Know how to figure out which search terms get you the right result
- Leetcode Easy to Medium

Coding Challenge Tips



hackerrank.com

AppleConnect BetaList CodePen Collect UI Events EECS Board FreeCodeCamp Full Stack Python Gym Internship Sheet Masters of Scale SourceForge Webcasts

User Dashboard | Make School \| Online... | How to Uninstall Node... | The Best Way to Instal... | Eta Kappa Nu (HKN),... | Course Outline - Goog... | Tutorials — Scotch | Getting Started with J... | Find the Duplicate Nu... | B.Hacking - the challe...

Booking.com B.Hacking - the challenge starts here! 49m : 50s to test end 2/4 Attempted John Yang

Sort hotel list

Given a set of hotel identifiers and some number of reviews from guests about a hotel, sort the hotel identifiers based on a list of keywords of interest (these will be given)
The criteria to sort the hotels should be how many times the interesting keywords are mentioned in any reviews about a given hotel.

Input:

There are 3 groups of input: keywords, hotel_ids, and reviews.

The first input contains keywords, as a single line of a space-separated words.
(If you use the pre-generated input code stubs, this will be the first string argument to your function. If you are parsing input yourself, this string of keywords will be the first line of input.)

The second line contains one integer `M`, which means the number of lines in the following group.
Then followed by `M` lines of integers, which mean the list of hotel IDs.

Then followed by one integer `N`, which means the number of lines in the following group.
Then followed by `N` lines of strings, which means the list of reviews. Each review is a single string, on a single line.

The list of hotel IDs and the list of reviews are parallel arrays. These two lists are always the same length (`N == M`). For example, if they are put into 2 arrays called `hotel_ids` and `reviews`, respectively, then `reviews[3]` is a review for hotel `hotel_id[3]`. (A hotel can have more than one review).

Output:
A list of hotel IDs. It should be sorted, in descending order, by how many mentions they have of the keywords across all of the reviews for a given hotel.
(If you are not using the pre-generated output code stubs, you should print each hotel ID on a separate line.)

Notes:
* The words to be find will **always** be single words like 'breakfast' or 'noise'. **Never** double words like 'swimming pool'.
* Hotel ID is a 4-byte integer.
* Matching should be case-insensitive.
* Dots and commas should be ignored.
* If a word appears in a review twice, it should count twice.
* If two hotels have the same number of mentions, they should be sorted in the output based on their ID, smallest ID first.
* In case one or more test cases time out, consider revisiting the runtime complexity of your algorithms.

Sample Input

```
import ↔;

public class Solution {


    /*
     * Complete the function below.
     */
    static int[] sort_hotels(String keywords, int[]
hotel_ids, String[] reviews) {
        // Map<Integer, Integer> map = new
HashMap<Integer, Integer>();
        List<Node> list = new ArrayList<Node>();
        String[] words = keywords.split(" ");
        for (int i = 0; i < reviews.length; i++) {
            int matches = countMatches(words,
reviews[i]);
            if (map.containsKey(hotel_ids[i])) {
                map.put(hotel_ids[i],
map.get(hotel_ids[i]) + matches);
            } else {
                map.put(hotel_ids[i], matches);
            }
        }

        PriorityQueue<Node> queue = new PriorityQueue
        for (Integer key : map.keySet()) {
            Node newNode = new Node(key, map.get(key)
            queue.add(newNode);
        }

        int[] output = new int[queue.size()];
        int index = 0;
        while (!queue.isEmpty()) {
            output[index] = queue.poll().key;
            index++;
        }
        return output;
    }

    static int countMatches(String[] a, String b) {
        int count = 0;
        for (int i = 0; i < a.length; i++) {
```

Phone Interview Tips

- Setting:
  - **Clear** Phone Connection
  - Relatively **quiet** space (home, reserved room)
  - Reliable Internet Connection
- Platforms: Codepair, Google Doc, Hackerrank
- Tips on Interacting with Interviewer
  - **Ask questions**, but don’t expect them to be answered and don’t get thrown off.
  - Take command, you drive the direction of the interview, not the interviewer.
  - DO NOT code right off the bat, outline your approach first, then start typing.

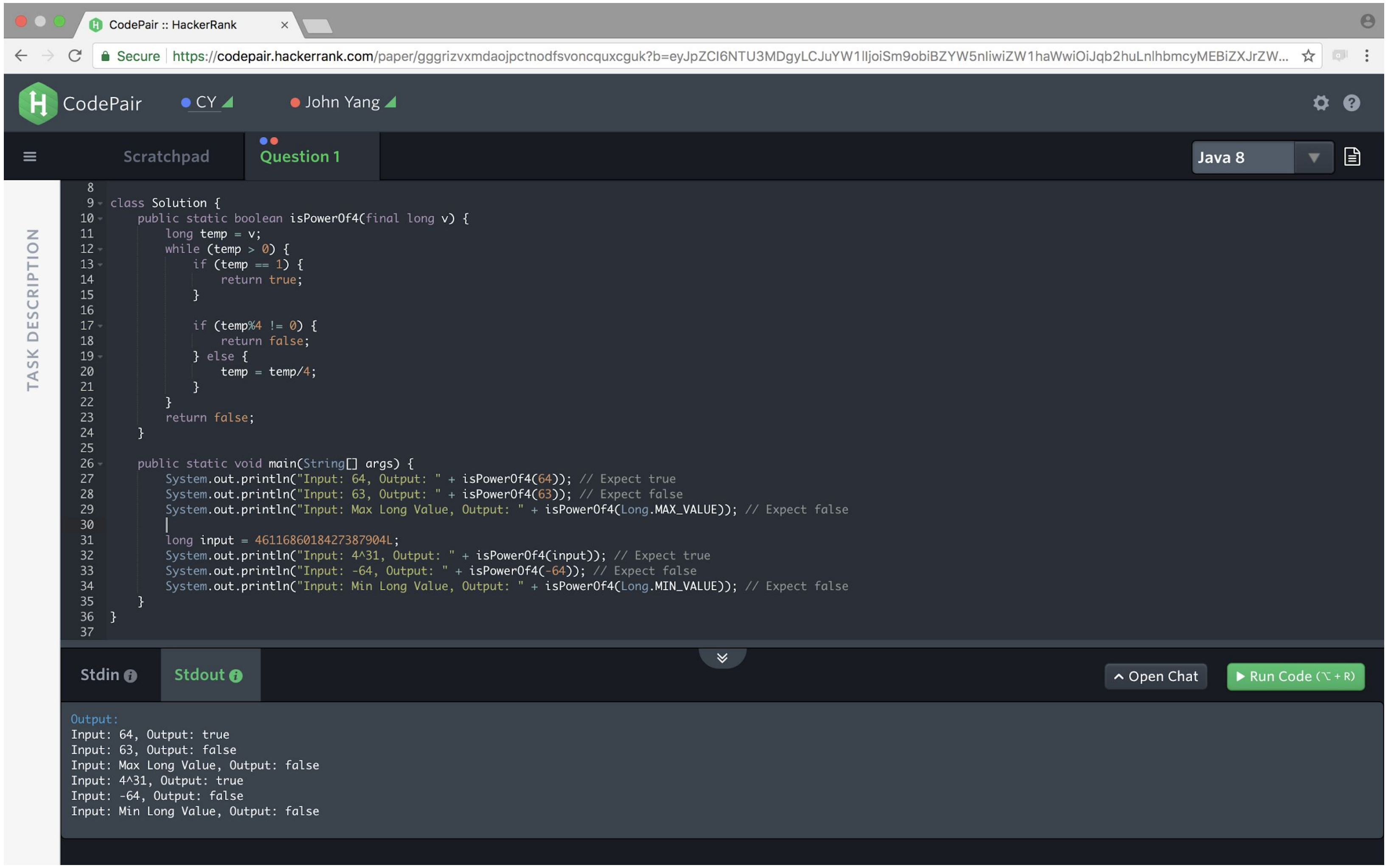
Phone Interview Tips

- Outline of Dialogue:
  1. Interviewer introduces themselves + role at the company (2-3 minutes)
  2. Introduce yourself (2-3 minutes, school + experience + behavioral question)
  3. Technical Portion (30-45 minutes)
  4. Questions for the interviewer (never a bad idea to be interested in them!) (3-5 minutes)
- Success in phone interviews varies more than coding challenges, but usually you have to arrive at the **correct** solution + answer any **optimization**, **testing**, or other technical questions correctly.

Phone Interview Tips

- Outline of Coding Challenge
    1. “Okay, give me a minute to read and understand the problem” (2-3 minutes)
    2. “I will talk through my approach, maybe we can discuss it, and if it sounds good, I can start coding?” (3-5 minutes)
    3. Implementation Time + Debugging and Refactoring (15-20 minutes)
    4. Either A. Interviewer gives you another problem, or B. asks you complexity and space analysis + test cases
- Leetcode Easy to Medium to Hard (Final Round)

Phone Interview Tips


CodePair :: HackerRank
Secure https://codepair.hackerrank.com/paper/gggrizvxmdaojpctnodfsvoncquxcguk?b=eyJpZCI6NTU3MDgyLCJuYW1lIjoiSm9obiBZYW5nIiwiZW1haWwiOiJqb2huLnlhbmcyMEBiZXJrZW...
CodePair
CY
John Yang
Scratchpad
Question 1
Java 8
TASK DESCRIPTION
Stdin
Stdout
Open Chat
Run Code (⌥ + R)

```java

class Solution {
    public static boolean isPowerOf4(final long v) {
        long temp = v;
        while (temp > 0) {
            if (temp == 1) {
                return true;
            }

            if (temp%4 != 0) {
                return false;
            } else {
                temp = temp/4;
            }
        }
        return false;
    }

    public static void main(String[] args) {
        System.out.println("Input: 64, Output: " + isPowerOf4(64)); // Expect true
        System.out.println("Input: 63, Output: " + isPowerOf4(63)); // Expect false
        System.out.println("Input: Max Long Value, Output: " + isPowerOf4(Long.MAX_VALUE)); // Expect false

        long input = 4611686018427387904L;
        System.out.println("Input: 4^31, Output: " + isPowerOf4(input)); // Expect true
        System.out.println("Input: -64, Output: " + isPowerOf4(-64)); // Expect false
        System.out.println("Input: Min Long Value, Output: " + isPowerOf4(Long.MIN_VALUE)); // Expect false
    }
}

```

```
Output:
Input: 64, Output: true
Input: 63, Output: false
Input: Max Long Value, Output: false
Input: 4^31, Output: true
Input: -64, Output: false
Input: Min Long Value, Output: false
```

Phone Interview Examples

- Optimization
  - Is there an approach that can solve this in faster runtime? (Improve time complexity)
  - Can you cut down on the amount of memory or operations you're using? (reduce space)
  - I've changed the problem slightly, what about your solution changes?
- Test Cases
  - Passes with regular input - Test whether your algorithm actually works.
  - Passes with errant input - Test whether code handles exceptions and edge cases properly

Onsite Tips

- Setting:
  1. At company headquarters!
  2. Interact with multiple employees + engineers
  3. Learn more about the company
  4. Filled with perks! This stage is fun + intense :D
- Typical Schedule
  - 4-6 Total Interviews:
    - 2-4 Technical Interviews
    - 2-3 Behavioral Interviews
  - Formats: Whiteboard, Collaborative Coding, OOP Design, Conversation about previous internships + experience

Onsite Tips

- What you can do
    1. Not business formal, but don’t dress sloppy either! Jeans / Khakis + T-Shirt / Polo + Hoodie / Sweater should be fine.
    2. Bring a backpack with copies of your resume, laptop, and pen + paper.
    3. Remember contact information! Super helpful for picking a team + staying in touch.
- Other notes
    - Hardest Interview Questions (Medium to Hard)
    - If it doesn’t work out this time, email your recruiter for re-interview next year!

Good luck!

http://saasbook.info