Code Jam - Study Jam

Josh Snider

3/29/2018

Intro

On 3/29/2018, Google Developer Group - Washington D.C. decided to hold a study session to prepare people for the Google Code Jam. The event was cosponsored by the DC Android and Kotlin Washington DC User Group meetups. These are my notes from the event. The basic schedule of the event was discussing three problems from previous Code Jams. The Capital One Android team was most of the volunteers at the event.

About Code Jam

The Code Jam schedule is here, but essentially there's a qualification round the weekend of April 6th. Then three online rounds and a final onsite round. I've participated in these code jams before, I think I've made it to either the second or third round. Distributed Code Jam requires you to have made it to round three before, so I must have made it that far at least once. Points are awarded by speed and accuracy of your solutions. There's a cutoff for points to get to the next round. Code Jam has very few restrictions on languages. Today's practice problems can be found at goo.gl/fbnUH9

Problem 1

Standing Ovation from the 2015 Qualification Round presented by Ahmad Ibrahim. You want everyone in the audience to clap, but every audience member has a shyness level that needs to be met before they will stand. What's the smallest number of people you can invite to get everyone to stand? I already had a solution from 2 years ago at https://github.com/jsnider3/Workspace/tree/master/Co
I made a new solution for this.

Problem 2

Revenge of the Pancakes from 2016 Qualification Round presented by Rokas Leskevicius. We want to take a stack of pancakes and make sure they are all facing up. Our only power is to flip the top N pancakes. How many times do you need to do this? You need to flip once for each time you have a '-' next to a '+' you then have to flip once more if that results in everything being face down.

Problem 3

Alphabet Cake from 2017 round 1A presented by Chuk-Yang Seng, we want to give each child a rectangular part of a cake covered in their initials. Cake can be divided unfairly, but must

divide the entire cake. If there are multiple solutions, we can output any. I think we can just do a greedy solution. My solution was to just expand letters left, up, right, and then down as much as possible. That passed the small input.