Problem Solving Workshop #1 Tech Interviews and Competitive Programming Meetup

February 13, 2016 https://www.meetup.com/tech-interviews-and-competitive-programming/

Instructor: Eugene Yarovoi (can be contacted through the group Meetup page above under Organizers)

More practice questions: leetcode.com, glassdoor.com, geeksforgeeks.org

Books: Elements of Programming Interviews, Cracking the Coding Interview

Have questions you want answered? Contact the instructor, or ask on Quora. You can post questions and follow the instructor and other people who write about algorithms.

Try to find optimized solutions, and provide a time and space complexity analysis with every solution for the algorithms questions.

Suppose you have a spreadsheet application. When a user clicks the mouse, you want to know which cell they're clicking on so that you can switch the focus to it. You get (x, y) mouse click coordinates as input, and you want to return the cell number clicked on.

Since solving for the row number when given the y-coordinate and solving for the column number when given the x-coordinate are analogous problems, you can just worry about computing the column number, given an x-coordinate.

Initially, you'll get an array *columnSizes* denoting the column sizes. For example, an input of [8, 2, 10, 10] means that there are 4 columns (numbered 0 through 3), with the 0th column having width 8 pixels, the 1st column having width 2, etc.

Then, you will receive as input a sequence of user operations. A user operation either has the form *Click(x)* or *Resize(index,newWidth)*. In the first case, the operation specifies an integer x, and you must print the column # of the column at pixel coordinate x. In the second case, you don't print anything, but you must update column # *index* to now have *newWidth*. This change may affect the result of subsequent *Click* operations.

The 0th column starts at the 0th pixel. With the earlier example array, if a *Click* occurs before any updates, if x is between 0 and 7 you should print 0, for x=8-9 print 1, for x=10-19 print 2, and for x=20-29 print 3.

(i) Give any correct solution to this problem. It can be inefficient. **Difficulty level: basic**

(ii) You want the *Click* operation to be very fast because it's a common operation. However, you don't care if *Resize* is very slow, since resizing columns is much less common. You can also take some time to pre-process the *columnSizes* array when you load the file. Give an algorithm that achieves *Click* in less than linear time with respect to *columnSizes.length*. **Difficulty: mid-tier company interview**

(iii) Accomplish both *Click* and *Resize* efficiently. **Difficulty: elite company interview**

(iv) There is now a third possible operation, *Insert(index,width)*. It inserts a new column numbered *index*, causing the column that was previously numbered *index* to now be numbered *index+1*, the column previously numbered *index+1* to be numbered *index+2*, etc. The new column has the specified *width* in pixels. **Difficulty: tad above (iii)**

(v) Instead of being what it was in (iv), the third operation is *ResizeRange(start,end,newWidth)*. It resizes all columns numbered between *start* and *end* to have *newWidth*. This must happen very efficiently even if the range covered is very large -- you cannot simply apply the solution from (iii) to every affected column. **(Difficulty: mid-level contest)**