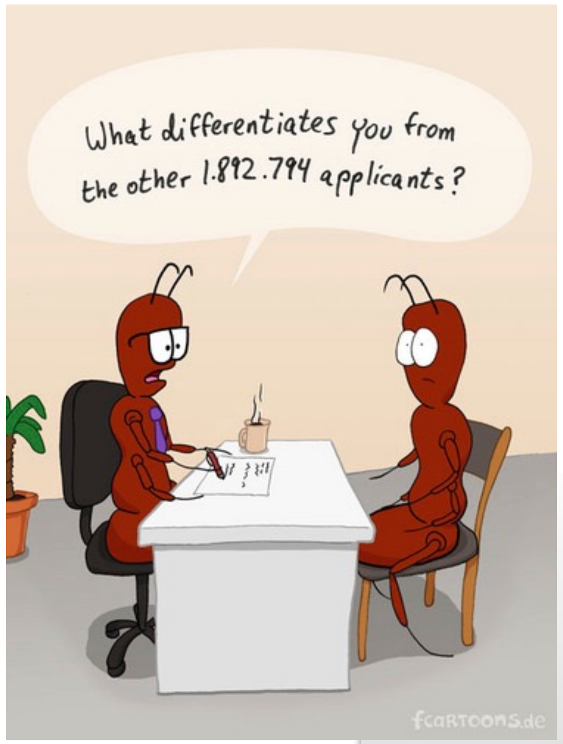
Information Session


What differentiates you from the other 1.892.794 applicants?
fcartoons.de

What makes you think you'd be a good fit with Have A Nice Day, Inc.?

Preparing for Developer Job Interviews

(sure, also at Google)

The World of Computing Some Years Ago

Woz!
I've just come up with a new algorithm for blah that runs in only O(n log*n / loglog n)!!!

You know Don,
"In The Real World"
we can't afford to spend time coding complex algorithms, we go for the easiest to code.

Donald Knuth
Stanford University

Steve Wozniak
Apple Inc. founder

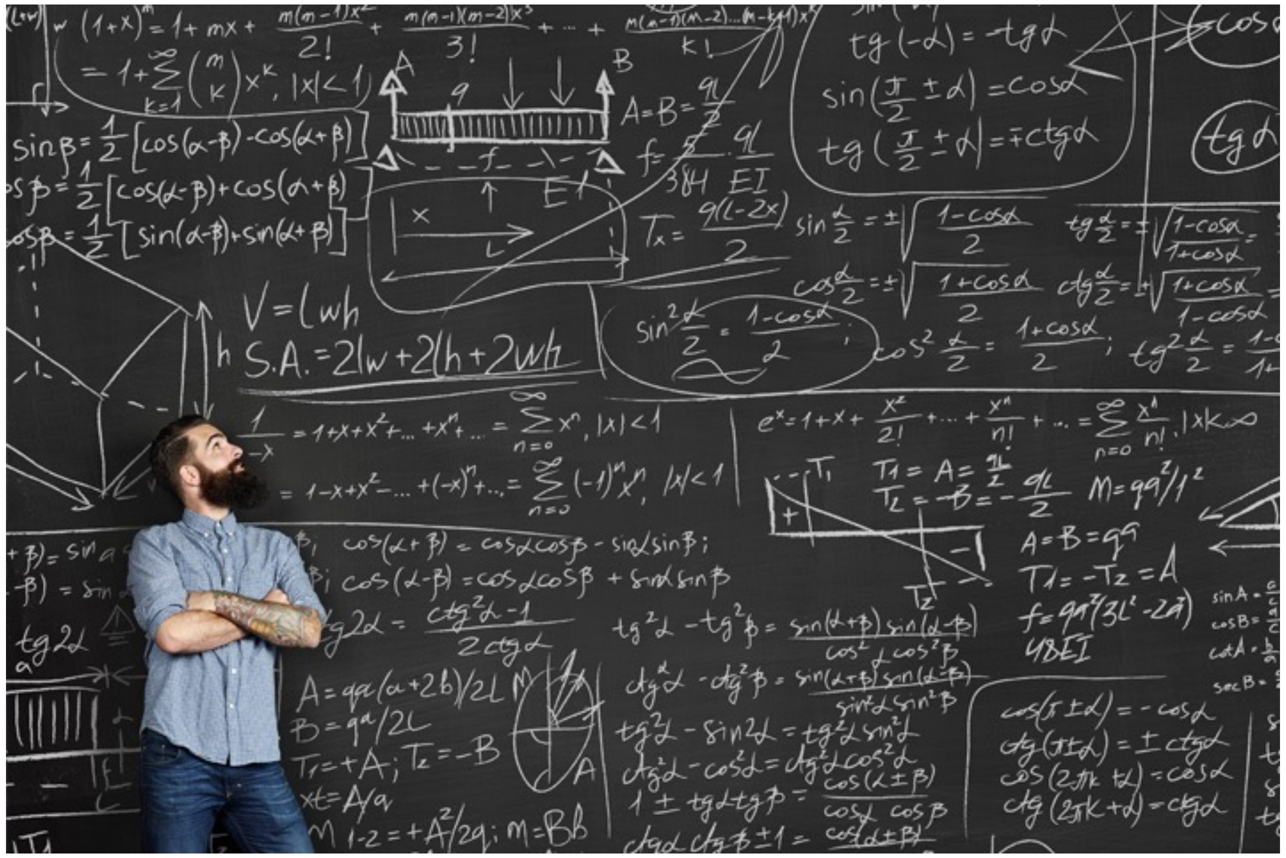
(FICTIONAL DIALOGUE)

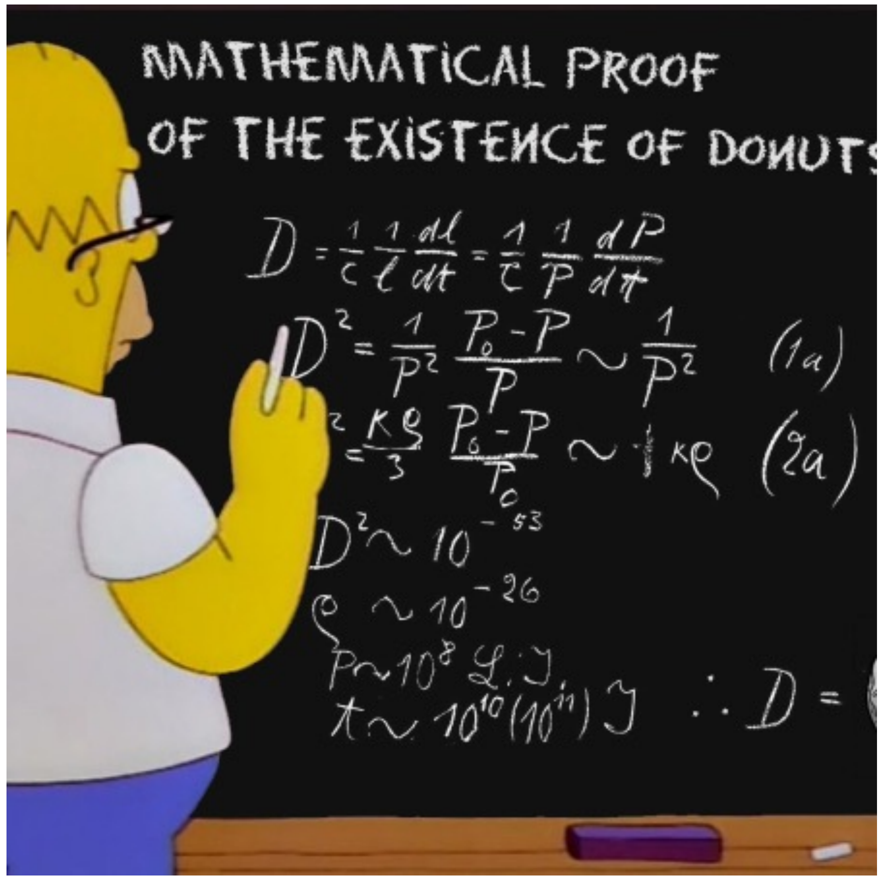
The World of Computing Today

FAST ALGORITHMS
ARE CRUCIAL !!

Algorithms and Data Structures are also "Real World"

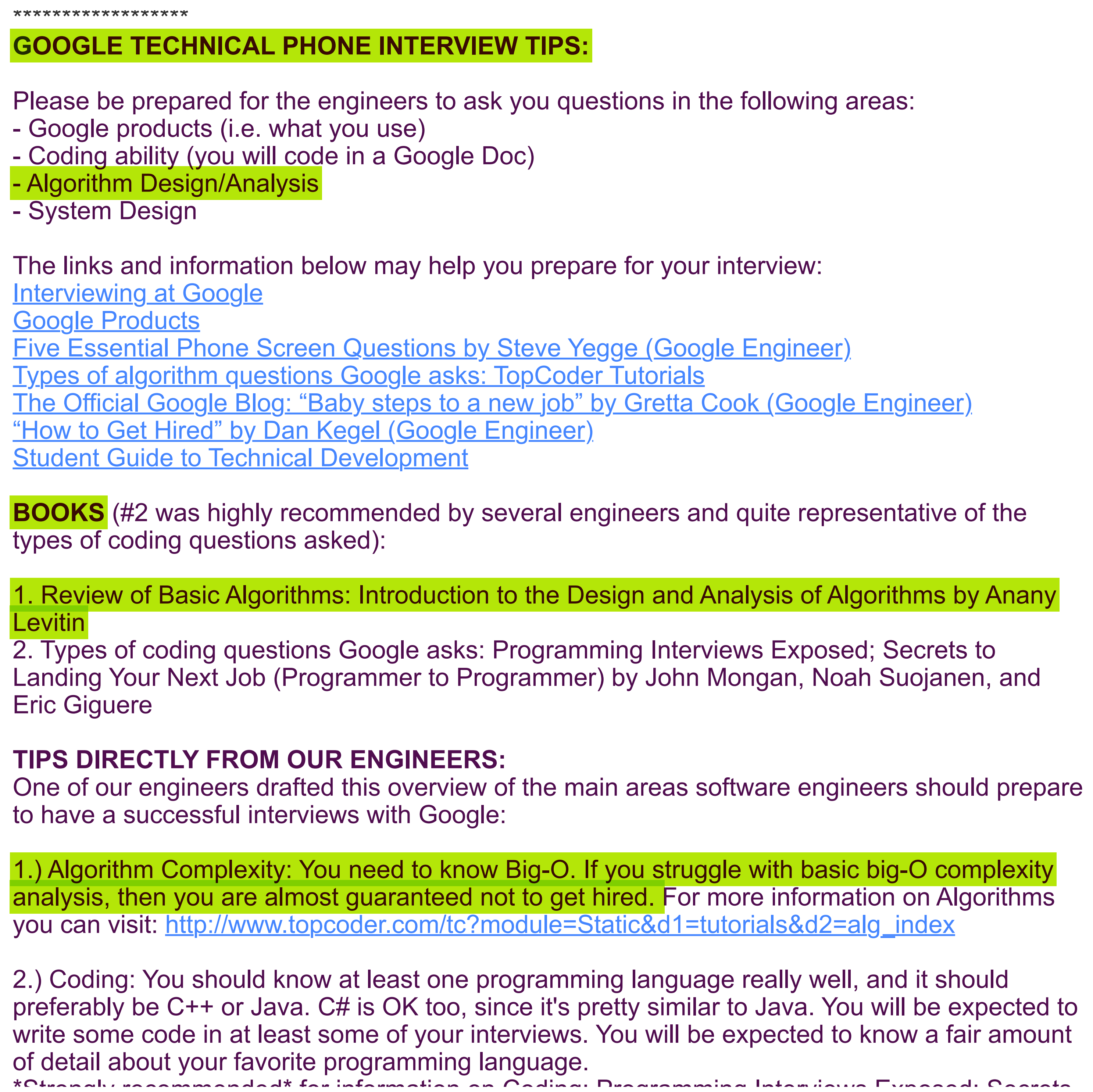
Back to the board





AND EMPLOYERS DO CARE ABOUT IT, A LOT!

Google Technical Phone Interview Tips



****

GOOGLE TECHNICAL PHONE INTERVIEW TIPS:

Please be prepared for the engineers to ask you questions in the following areas:
- Google products (i.e. what you use)
- Coding ability (you will code in a Google Doc)
- Algorithm Design/Analysis
- System Design

The links and information below may help you prepare for your interview:
Interviewing at Google
Google Products
Five Essential Phone Screen Questions by Steve Yegge (Google Engineer)
Types of algorithm questions Google asks: TopCoder Tutorials
The Official Google Blog: “Baby steps to a new job” by Gretta Cook (Google Engineer)
“How to Get Hired” by Dan Kegel (Google Engineer)
Student Guide to Technical Development

BOOKS (#2 was highly recommended by several engineers and quite representative of the types of coding questions asked):

1. Review of Basic Algorithms: Introduction to the Design and Analysis of Algorithms by Anany Levitin
2. Types of coding questions Google asks: Programming Interviews Exposed; Secrets to Landing Your Next Job (Programmer to Programmer) by John Mongan, Noah Suojanen, and Eric Giguere

TIPS DIRECTLY FROM OUR ENGINEERS:
One of our engineers drafted this overview of the main areas software engineers should prepare to have a successful interviews with Google:

1.) Algorithm Complexity: You need to know Big-O. If you struggle with basic big-O complexity analysis, then you are almost guaranteed not to get hired. For more information on Algorithms you can visit: http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=alg_index

2.) Coding: You should know at least one programming language really well, and it should preferably be C++ or Java. C# is OK too, since it's pretty similar to Java. You will be expected to write some code in at least some of your interviews. You will be expected to know a fair amount of detail about your favorite programming language.
Strongly recommended for information on Coding: Programming Interviews Exposed; Secrets

Google Technical Phone Interview Tips

absolutely should know how they work. Be able to implement one using only arrays in your favorite language, in about the space of one interview.

6.) Trees: Know about trees; basic tree construction, traversal and manipulation algorithms. Familiarize yourself with binary trees, n-ary trees, and trie-trees. Be familiar with at least one type of balanced binary tree, whether it's a red/black tree, a splay tree or an AVL tree, and know how it's implemented. Understand tree traversal algorithms: BFS and DFS, and know the difference between inorder, postorder and preorder.

7.) Graphs: Graphs are really important at Google. There are 3 basic ways to represent a graph in memory (objects and pointers, matrix, and adjacency list); familiarize yourself with each representation and its pros & cons. You should know the basic graph traversal algorithms: breadth-first search and depth-first search. Know their computational complexity, their tradeoffs, and how to implement them in real code. If you get a chance, try to study up on fancier algorithms, such as Dijkstra and A*.

8.) Other data structures: You should study up on as many other data structures and algorithms as possible. You should especially know about the most famous classes of NP-complete problems, such as traveling salesman and the knapsack problem, and be able to recognize them when an interviewer asks you them in disguise. Find out what NP-complete means.

9.) Mathematics: Some interviewers ask basic discrete math questions. This is more prevalent at Google than at other companies because we are surrounded by counting problems, probability problems, and other Discrete Math 101 situations. Spend some time before the interview refreshing your memory on (or teaching yourself) the essentials of combinatorics and probability. You should be familiar with n-choose-k problems and their ilk – the more the better.

10.) Operating Systems: Know about processes, threads and concurrency issues. Know about locks and mutexes and semaphores and monitors and how they work. Know about deadlock and livelock and how to avoid them. Know what resources a processes needs, and a thread needs, and how context switching works, and how it's initiated by the operating system and underlying hardware. Know a little about scheduling. The world is rapidly moving towards multi-core, so know the fundamentals of "modern" concurrency constructs.

For information on System Design:
http://research.google.com/pubs/DistributedSystemsandParallelComputing.html

11.) Also, you can review some of our research publications: http://research.google.com/pubs/papers.html
and paper on Google's Hybrid Approach to Research:
http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/38149.pdf

12.) Exercise your coding skills on TopCoder. We've heard from engineers that it's great practice for a Google technical interview!

Google Technical In-person Interview Tips

"How to: Work at Google" — Example Coding/Engineering Interview
24 min video .

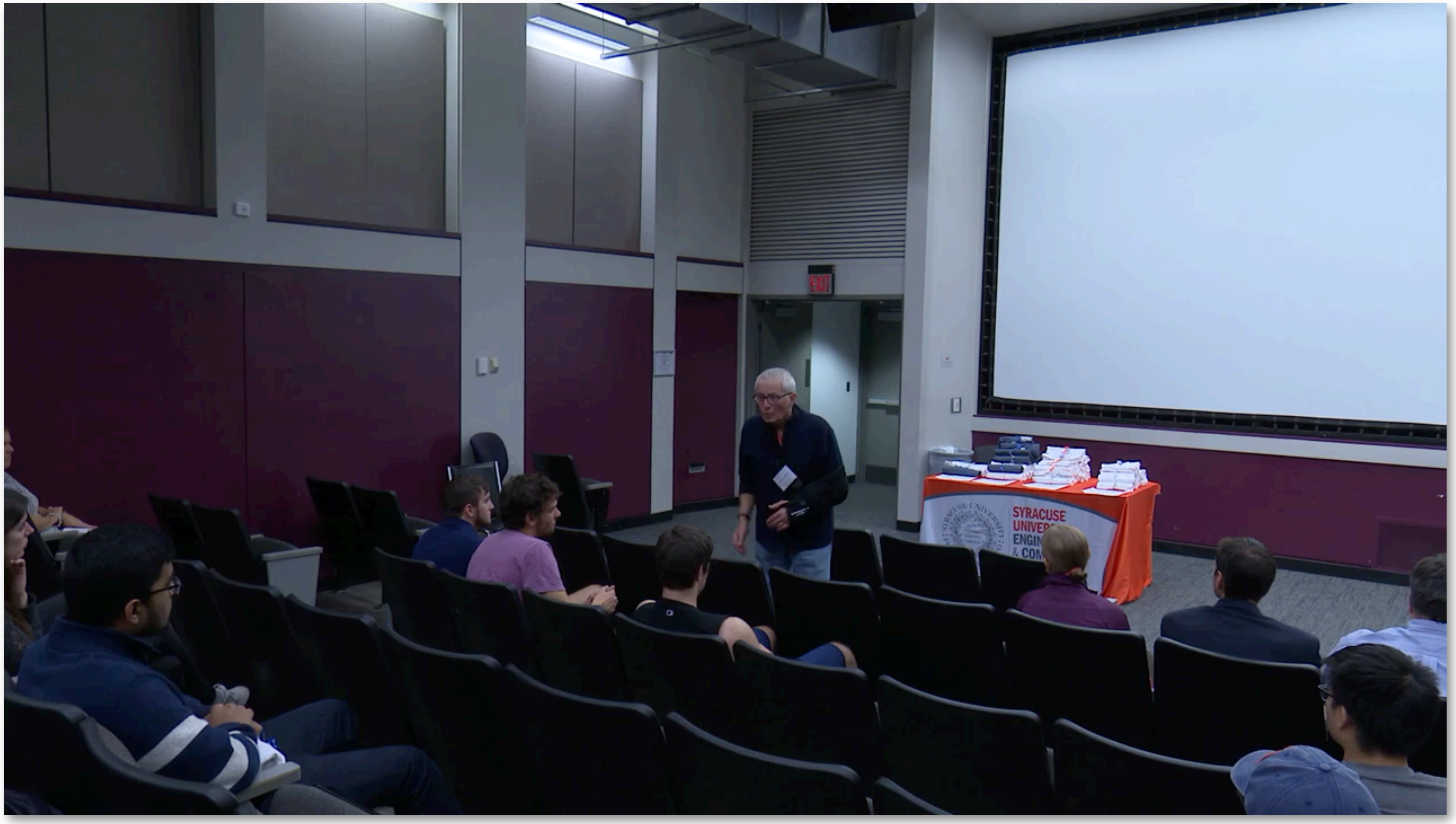
How are we going to be``measured" in industry?

…by how we create new knowledge.

To create new algorithms,
The key is to understand how the current were designed.

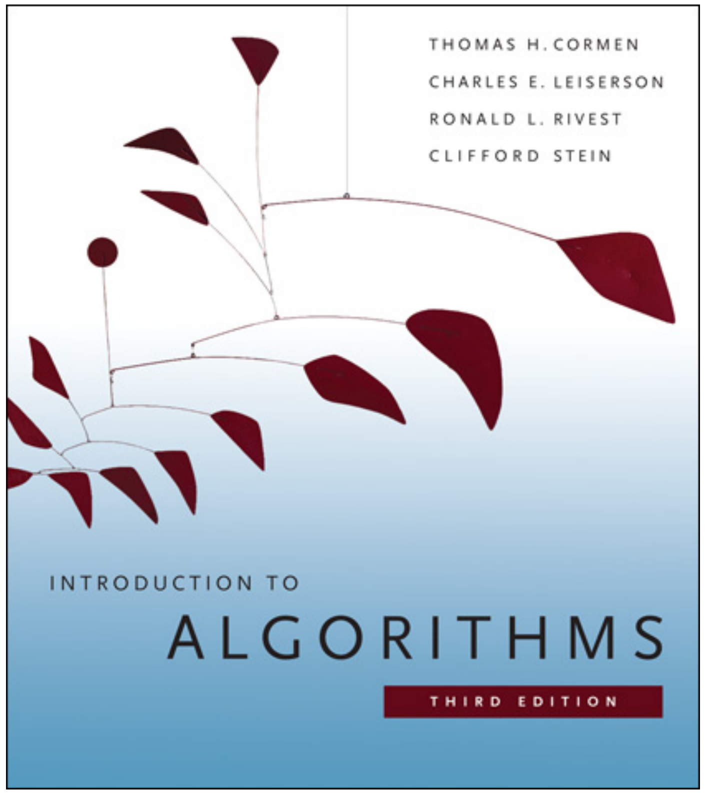
Don't settle to be an expert user of the hottest tool,
be an expert designer of the next hottest tool!!

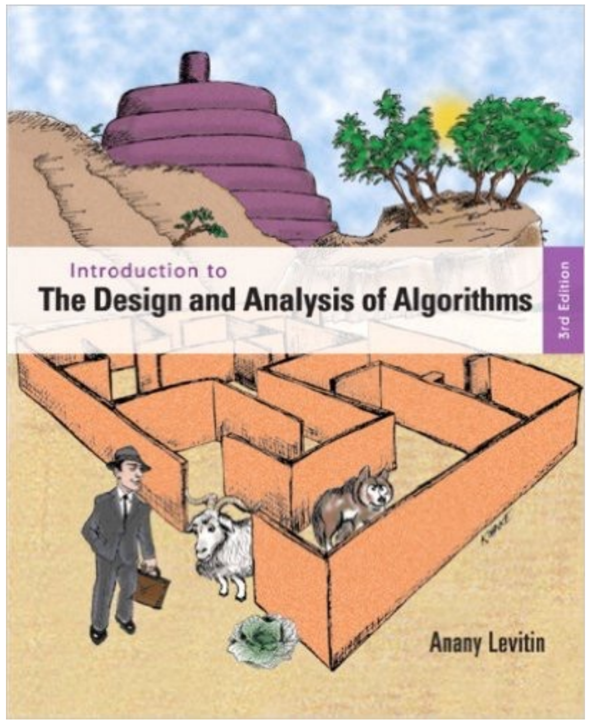
How are we going to be``measured'' in industry?

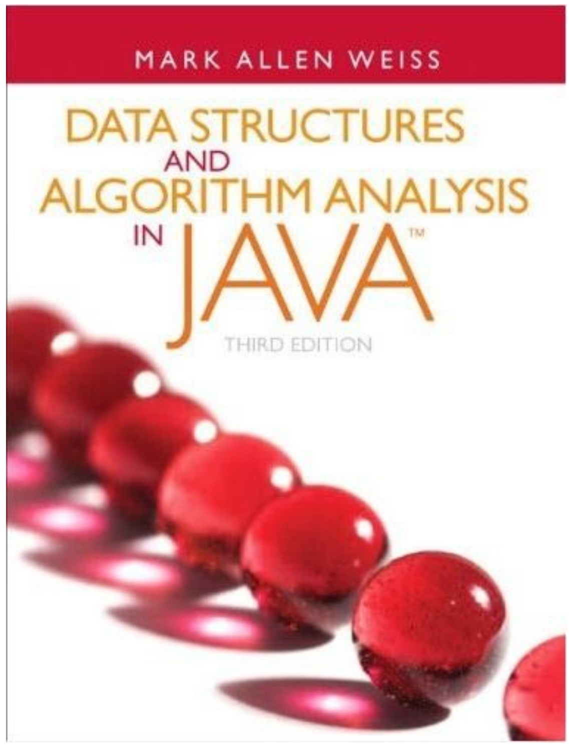


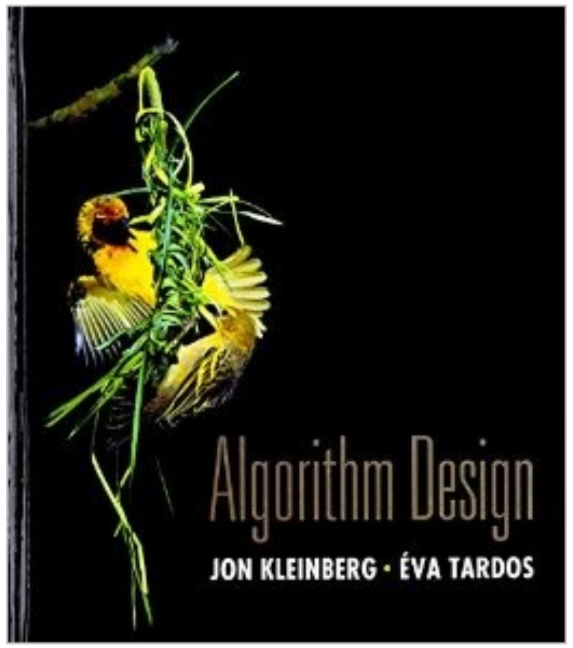
Nick Donofrio, ExecVP Innovation & Tech at IBM 5 min video.

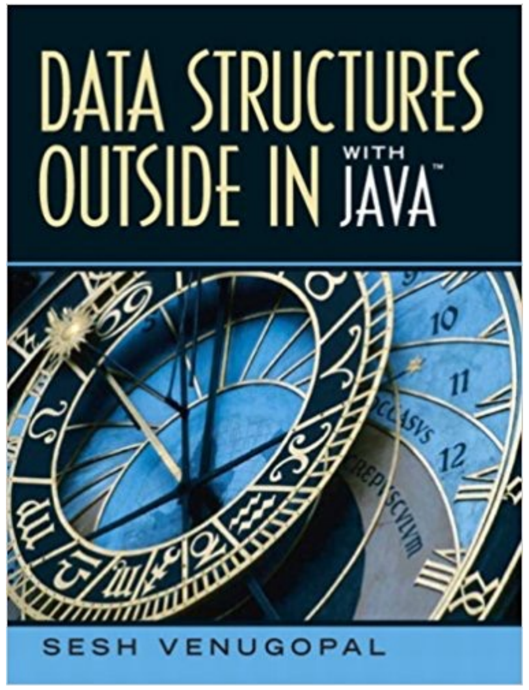
Books you can use to catch-up











Videos you can use to catch-up

Erik Demaine (MIT) Fall 2011
Charles Leiserson (MIT) Fall 2005
Robert Sedgewick (Princeton) part I
Robert Sedgewick (Princeton) part II
Tim Roughgarden (Stanford)
Tom Leighton (MIT) Math for CS

Questions?