Disclaimer

I’m here today to share my own views and opinions, which don’t necessarily reflect the views and opinions of my past or present employers.
Who am I?

@rcdickerson

github.com/rcdickerson
The Other Side Of The Table

What is the interviewer trying to accomplish?

- Find a good employee.
- “Smart and gets things done.” ~ Joel Spolsky
- Someone who will “raise the bar”.
- Avoid a hiring mistake.
- Figure out where you fit on some competency scale.
# The Programmer Competency Matrix

<table>
<thead>
<tr>
<th>Computer Science</th>
<th>$2^n$ (Level 0)</th>
<th>$n^2$ (Level 1)</th>
<th>$n$ (Level 2)</th>
<th>$\log(n)$ (Level 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>data structures</strong></td>
<td>Doesn’t know the difference between Array and LinkedList</td>
<td>Able to explain and use Arrays, LinkedLists, Dictionaries etc in practical programming tasks</td>
<td>Knows space and time tradeoffs of the basic data structures, Arrays vs LinkedLists, Able to explain how hashtables can be implemented and can handle collisions, Priority queues and ways to implement them etc.</td>
<td>Knowledge of advanced data structures like B-trees, binomial and fibonacci heaps, AVL/Red Black trees, Splay Trees, Skip Lists, tries etc.</td>
</tr>
<tr>
<td><strong>algorithms</strong></td>
<td>Unable to find the average of numbers in an array (it’s hard to believe but I’ve interviewed such candidates)</td>
<td>Basic sorting, searching and data structure traversal and retrieval algorithms</td>
<td>Tree, Graph, simple greedy and divide and conquer algorithms, is able to understand the relevance of the levels of this matrix.</td>
<td>Able to recognize and code dynamic programming solutions, good knowledge of graph algorithms, good knowledge of numerical computation algorithms, able to identify NP problems etc.</td>
</tr>
<tr>
<td><strong>systems programming</strong></td>
<td>Doesn’t know what a compiler, linker or interpreter is</td>
<td>Basic understanding of compilers, linker and interpreters. Understands what assembly code is and how things work at the hardware level. Some knowledge of virtual memory and paging.</td>
<td>Understands kernel mode vs. user mode, multi-threading, synchronization primitives and how they’re implemented, able to read assembly code. Understands how networks work, understanding of network protocols and socket level programming.</td>
<td>Understands the entire programming stack, hardware (CPU + Memory + Cache + Interrupts + microcode), binary code, assembly, static and dynamic linking, compilation, interpretation, JIT compilation, garbage collection, heap, stack, memory addressing...</td>
</tr>
</tbody>
</table>


+ Use this type of thing as a guideline
+ I’m not saying interviewers literally use this matrix, although this is a good one to think about
Your Attitude As An Interviewee

- Your interviewer is setting you up to **knock one out of the park!**
- ... so **don’t be timid** or stick to minimal answers
- ... but **don’t pretend to know things you don’t**
- Be **relaxed, positive, and passionate**
Identifying Company Culture

Painting with a broad brush...

Traditional Companies

- Heavy on verbal interviews
- Compensation is mostly salary
- Nine-to-five

Start-up Companies

- Heavy on coding interviews
- Compensation is a combination of salary and equity
- Longer and more flexible hours
Cultural Considerations
What type of company do you want to work for?

- **Challenge Level**
  - You will get to (have to?) work with the type of people who get hired here.

- **Growth Opportunity**

- **Work / Life Balance**
The Basic Hiring Funnel

1. Resume screen
2. Phone screen
3. On-site interview
4. Offer
The Résumé

- You should be doing things outside of your coursework
  - Side projects
  - Open source projects
- Read Steve Yegge’s “Ten Tips for a (Slightly) Less Awful Resume”

Interview Techniques You Are Likely to Encounter

- Highly dependent on company culture.
- We’ll talk about four of the most common:
  - Verbal interviews
  - Written tests
  - Whiteboard coding
  - Pair programming

Thursday, October 17, 13
Verbal Interviews

Purpose

- Assess cultural fit
- Assess communication skills

You Should

- Have a dialog, ask questions
- Not be a jerk

Common Pitfalls

- Being timid
- Not being prepared to talk about past projects
Written Tests

Purpose
- Assess basic CS knowledge

You Should
- Review basic CS concepts before the interview

Common Pitfalls
- Pretending to know things you don’t
- Poor time management

Thursday, October 17, 13
Whiteboard Coding

Purpose

- Assess high-level design and problem solving ability
- Assess basic algorithm knowledge

You Should

- Brush up on OOP and basic algorithms before the interview

Common Pitfalls

- Becoming a deer in the headlights
- Getting tripped up on syntax concerns
Coding Interviews

Purpose
- Assessing actual coding behavior and ability

You Should
- Explain yourself and write tests as you go
- ABC (Always Be Coding)

Common Pitfalls
- Hesitating to jump in and write code
- Not backing up to fix design mistakes

Thursday, October 17, 13
+ ABC courtesy of David Bytow. Google it for a good read.
Topics To Be Ready For

- Data Structures
  - Arrays
  - Strings
  - Linked Lists
  - Trees
  - Graphs
- Recursion
- OOP

Thursday, October 17, 13

- Cracking the Coding Interview – Gayle Leakmann McDowell
- Programming Interviews Exposed – Mongan, Suojanen, and Giguere
Red Flags

Interview anti-patterns

- Brainteasers
- Interview processes that are all verbal
- Interview processes where you barely talk to other engineers
- Cliché questions
- A disorganized interview process
The Day Before the Interview

- Work through practice problems.
- Make sure you know about the company, its products, and its culture.
- Get some sleep.
The Day Of The Interview

- Dress appropriately
  - At or one step above the norm for the office
  - If in doubt, it’s OK to ask the recruiter what to wear
- Warm up with some more practice problems
- Show up on time.
- No, for real. Be on time.
Thinking About Compensation

- Salary
- Stock Options
- Retirement
- Health and Other Benefits
Square’s Interview Process
Questions?