CRACKING THE CODING SKILLS

Created By Gayle Laakmann McDowell

Best Conceivable Runtime (BCR)

BCR is the runtime you *know* you can't beat. For example, if asked to compute the intersection of two sets, you know you can't beat $O(|A|+|B|)$.

5 Approaches

- **BUD:** Look for bottlenecks, unnecessary work, duplicated work.
- **DIY:** Do It Yourself
- **Simplify & Generalize:** Solve a simpler version.
- **Base Case & Build:** Solve for the base cases then build from there.
- **Data Structure Brainstorm:** Try various data structures.

1 Listen

Pay very close attention to any info in the problem description. You probably need it all for an optimal algorithm.

BUD Optimization

Bottlenecks

Unnecessary Work

Duplicated Work

7 Test

Test in this order:

1. Conceptual test. Walk through your code like you would for a detailed code review.
2. Unusual or non-standard code.
3. Hot spots, like arithmetic and null nodes.
4. Small test cases. It's much faster than a big test case and just as effective.
5. Special cases and edge cases.

And when you find bugs, **fix them carefully!**

6 Implement

Your goal is to **write beautiful code**. Modularize your code from the beginning and refactor to clean up anything that isn't beautiful.

2 Example

Most examples are too small or are special cases. **Debug your example.** Is there any way it's a special case? Is it big enough?

3 Brute Force

Get a brute-force solution as soon as possible. Don't worry about developing an efficient algorithm yet. State a naive algorithm and its runtime, then optimize from there. Don't code yet though!

4 Optimize

Walk through your brute force with **BUD optimization** or try some of these ideas:

- Look for any unused info. You usually need all the information in a problem.
- Solve it manually on an example, then reverse engineer your thought process. How did you solve it?
- Solve it "incorrectly" and then think about why the algorithm fails. Can you fix those issues?
- Make a time vs. space tradeoff. Hash tables are especially useful!

5 Walk Through

Now that you have an optimal solution, **walk through your approach in detail**. Make sure you understand each detail before you start coding.

What You Need To Know

1 **Data Structures:** Hash Tables, Linked Lists, Stacks, Queues, Trees, Tries, Graphs, Vectors, Heaps.

2 **Algorithms:** Quick Sort, Merge Sort, Binary Search, Breadth-First Search, Depth-First Search.

3 **Concepts:** Big-O Time, Big-O Space, Recursion & Memoization, Probability, Bit Manipulation.

Exercises:

- Implement data structures & algorithms from scratch.
- Prove to yourself the runtime of the major algorithms.

Books by Gayle

Do not...

- **Do not** ignore information given. Info is there for a reason.
- **Do not** try to solve problems in your head. Use an example!
- **Do not** push through code when confused. Stop and think!
- **Do not** dive into code without interviewer "sign off."

© CareerCup.com

CRACKING THE SOFT SKILLS
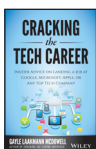
Created By Gayle Laakmann McDowell

Must Knows

- "So, tell me a bit about yourself…"
- "Why do you want to work here?"
- "Why should we hire you?"
- "Why are you leaving your current job?"
- "Where do you see yourself in 5 years?"
- "What do you do outside of work?"
- "What are your strengths and weaknesses?"

Your Two-Minute Pitch

"I am a _____ at _____."

"In college, I studied ___ at ____."

"Then I worked for … where I …."

"Then I worked for … where I …."

"In my current job, I've accomplished…."

"Also, outside of work, I …."

Preparing for Behavioral Questions

Create Resume Grid ⇒ Pick Five Key Stories ⇒ Diagram Your Stories

Themes	Job 1	Job 2
Leadership & Influence	story	
Mistakes & Failures		story
Challenges	story	
Teamwork	story	story
Successes		story

Coder? Add: Bugs, Architecture, Optimization, Scaling.

Stories	Story 1	Story 2
THEME(S)	e.g., Leadership, Challenge, …	
Nugget	"Sure, let me tell you about when I …"	
Situation	Only the basics needed	
Action(s)	Expand here!	
Result	Prove it!	
THE POINT	What does it say about you?	

Structured Answers

N — **Nugget.** "Sure, let me tell you about the time that I …" This focuses you and your interviewer on what you're about to say.

S — **Situation.** Explain just basics. The interviewer only needs enough details to understand what you did. Most people spend too much time here.

A — **Action(s).** Detail what actions you took. "First, I …. Then, I …. And finally, I …." This is where you should spend most of your time. Speak in bullets!
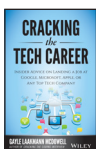
R — **Result.** Succinctly explain the result of your efforts were. Prove that the impact was good with numbers or a clear success metric.

Books by Gayle

Check Your Stories

- Are they substantial?
- Are they understandable?
- Have you explained *why* you did it this way?
- What do they say about you?
- Are they really about you (not your team)?
- Have you covered all the themes?
- Can you answer, "What would you do differently?"?

Questions For Your Interviewer

- Things you want to know.
- Things that show passion/interest.
- Things that show skills.