QCS Technical Skills Challenge

Week 1: Technical Interview Basics

Part 1: FizzBuzz
(A case study for answering Technical Questions)

What is FizzBuzz?

That’s pretty easy, why are we looking at it?

How to approach a problem...

0. Ask Questions. Lots of questions.

Even in this very constrained situation there's still a lot of clarification that we could ask for. Do we need to factor in decimal points? Will it always need to be printed FizzBuzz or will BuzzFizz do? Is extensibility an issue (Do we need to make it easy to add further modifiers)?

Don't be scared to talk to your interviewer. Chances are they want you to succeed as much as you do. Asking questions shows that you can anticipate future problems and can identify when you don't know something.

0.5. Explain everything you’re doing

Your fantastic design skills are wasted if you’re staring at a whiteboard for twenty minutes in silence. Talk through everything. What you’re thinking, what your assumptions are, why you’re thinking what you’re thinking.

Just like at GCSE you get points for working out. Even a dead wrong answer can reflect well on you if you can show and ability to think through a problem, identify flaws in your thinking, ask questions and use the answers...

1. Try to do it yourself

- Answer the question as if it was not for an algorithm. How do you, a person, approach this problem? How would you approach finding an answer if you didn't know it off the top of your head
- Is 5674 FizzBuzz?
- What about 5676?
- Try to write down the thought process you go through. What could a computer do? What could it not do?

2. Try to do it as if you were a computer

Think about what your computer can’t do that comes naturally to you. For us it is instinct that if we’re told to say “Fizz” and “Buzz”, it’ll come out as “FizzBuzz”. For a computer this needs to be explicitly defined.

What can a computer do that you can’t? It’s as easy for a computer to figure out if 3 is a mutiple of 3 as it is for it to figure out if 244 is.

3. Use Analogies, Use paper

For FizzBuzz this doesn’t apply so well but if you have a look at some of the other questions on the sheet you might start to see where physical analogies can help.

Your brain is designed to think about things, not data. Exploit that. Think about how you would solve this if you were calling it out. How would you solve a problem in another context?

Let’s try and write a first draft...

Can this be improved?

4. Think about ways to iterate your ideas

- What problem are you trying to solve?
- Is any processing being replicated?
- Can any parts of your code be pulled out into a named function? Can those parts be expressed differently?
- Does this problem mirror any that you have seen before in other contexts?

5. Think about how you'd test your success

If it's a long technical challenge or something, write tests so you can quickly see if you've succeeded

If you're in an interview, think about