Me: All right, this is Edward Auttonberry interviewing <JS>. I’m going to be giving you an alias by the way, like…

<JS>: Okay.

Me: I’m not actually going to write your name in the…

<JS>: I appreciate that.

Me: Well… that, uh… It’s required.

<JS>: Mm.

Me: I’m going to name you JavaScript.

<JS>: Mm. Nice.

Me: Because JS…

<JS>: Gotcha. Gotcha.

Me: Anyway.

<JS>: That’s actually pretty funny.

[Both laughing]

Me: I’m going to call… I’m going to call <DocX> DocX…

<JS>: Nice. Nice.

Me: Because like Microsoft Word docx.

<JS>: Mm. There you go.

Me: All right.

[Clap]

Me: Well then, what is algorithmic complexity, and what are complexity classes?

<JS>: Algorithmic complexity is… [Long pause] I would say… the… I want- I want to be careful about exactly the way I answer that question because it has several, um… it has several possible forms it could take. Uh, most people consider, like the time or… or memory complexity, as in the amount of time required for each step of the algorithm and each iteration of the algorithm and the amount of space required for each copy that the algorithm might perform on the data-that kind of thing. So, whether the algorithm is too costly for the amount of resources the computer can actually handle. And did you say algorithmic, or complexity…

Me: Complexity classes.

<JS>: … classes. That’s probably… I-I don’t really remember exactly that name, but I would guess it would be the things that I was talking about with time and memory and, um… uh what was the other thing? Uh, readability. The uh… um… The complexity in terms of how complex your code is and how difficult it is to grasp.

Me: Conceptual complexity.

<JS>: That! Yes, that one. Yep. That’s my guess for that one.

Me: Okay. So, just to steer you in the right direction, complexity classes are… are things we talked about like… you know we started with big-O some runtime, so…

<JS>: Oh, so it…

Me: Constant, polynomial…

<JS>: That’s what you meant by those things. Yes, I mean…

Me: And also, there is, you know, NP, NP-Complete. Those are all, like…

<JS>: Gotcha.

Me: Theoretical complexity classes.

<JS>: And I can name you all of those. I just didn’t know they were called complexity classes.

Me: Yeah, <DocX> was kind of confused too…

<JS>: Yeah.

Me: But that’s just what they’re called on the internet.

<JS>: Nice, nice.

Me: All right. So, what is an example of an NP problem? What is-What is your best example of an NP problem? Just an example.

[Heavy breathing by <JS>]

<JS>: Best example of an NP problem… We talked about the… complexity, we talked about the… potential for them to become polynomial time, but the only-the only one that we really went over in class that was explicitly exponential was… um, the one where you iterated over… um, a certain number of bits, and for each bit, you could have… um, any number of digits, or something like that, and basically it became…

Me: So, permutations of binary numbers.

<JS>: Right. Exactly. And so… that was explicitly exponential though. I don’t think we actually, in class, went over an example of an NP problem. We were told about… what an NP problem was, the implications of an NP problem, how it could become a polynomial problem, and… um, and how potentially… solving it could change the landscape of programming and encryption and pretty much everything in computation.

Me: Mmhmm.

<JS>: So…

Me: So, would you like me to give you an example?

<JS>: Please. Yes.

Me: So, one of them – I think the first example we were given – was the knight’s tour.

<JS>: Mm.

Me: As an NP problem.

<JS>: It is!

Me: Yeah.

<JS>: Okay.

Me: There was…

<JS>: Okay.

Me: The traveling salesman.

<JS>: Yeah! That one.

Me: Some other… graph theory…

<JS>: Okay. Yeah. Okay, Okay. Uh, the traveling salesman one was starting in one location and figuring out how to visit each location around the graph or map or whatever…

Me: Mmhmm, of a weighted graph.

<JS>: In-In the quickest… mm… In the quickest manner. Right? And knight’s tour is the same thing but taking a knight around a chess board. I didn’t know those were NP problems. That’s - That’s cool.

Me: Mmhmm.

[Mouth smack]

Me: What makes an NP problem different from an NP-Complete problem?

<JS>: An NP-Complete problem would be [pause] the harder more [pause] desirable to be solved of the NP problems because they are the ones that can be transformed… sorry. They’re the ones that NP problems can be transformed into at polynomial time, which means if an NP problem is solved in polynomial time, then all of the NP problems can be transformed into that at polynomial time, and become polynomial time themselves, just, like, double polynomial time…

Me: Mmhmm.

<JS>: Which is still polynomial.

Me: So, like you just said, what happens if an NP-class problem is proven to belong to either polynomial time or exponential time, in those spaces?

<JS>: Exactly. So, when an NP-Problem is solved in polynomial time, we will still celebrate, but I think it has to be one of the NP-Complete being solved in polynomial time that will allow all of the other NP problems to be polynomial. I think it’s... yeah.

Me: Okay. In what kind of situation would an understanding or application of this concept be useful?

<JS>: In any theoretical computing environment ever. Essentially, um… machine learning, cryptography, um… anything that involves iterating over a large amount of data in the fastest way possible would benefit from having [pause] accomplished what we talked about in the last question…

Me: Mmhmm.

<JS>: Because, the way cryptography works now, were basically making it too hard for our current computing systems to brute-force the answer to… like an – like a password. But if you had solved any one of the NP-Complete problems, brute-forcing becomes a polynomial time problem, and you’d be able to finish it in minutes. Even the biggest of passwords that we have. It would change… everything, just about – anything that’s on the theoretical end of modern computing. AI, gosh. We would do crazy things.

Me: Okay. Last question – In your opinion, do you think that this is an important topic for computer scientists/software engineers/cyber-security experts to understand? This is a two-part question by the way, so that’s the first question.

<JS>: Yes.

Me: Yes? Okay.

<JS>: Very much. Very, very, very much.

Me: Do you care to explain just a little bit?

<JS>: Okay, um… continuing kind of what I was answering in the last one, um… the more people, and the more eyes we have looking at these problems, the more potential we have for someone being able to think outside the box and solve one of these things, and if that happens some people may not benefit but, society and computing as a whole I think would benefit.

Me: Alright.

<JS>: And the second half of the question?

Me: What about people who do not work in those fields?

[Long pause]

<JS>: I think they should have a rudimentary understanding of it. Like they don’t have to actually participate in trying to solve the algorithms, but as long as they understand that those algorithms are… uh – that they have the potential to change even the fields that they’re working in that have nothing to do with cryptography or AI or any of that stuff, then they can at least appreciate the people who are working on it…

Me: Mmhmm

<JS>: And they can follow along at meetings that have to do with what could pot – because everything would change. You would have to have contingencies, businesses would have to prepare for that kind of thing, and, for the people who don’t understand the implications, they would drastically underestimate what would need to change, and so if we could teach it in a um… uh... layman, simple-to-understand kind of way then it would make the jobs of the people who are working on it far easier.

Me: So, what about like, school – high school students, secondary-school students, or maybe even like primary school students?

<JS>: If you can teach them what an algorithm is, then you can teach this to them. How’s that? If it is possible for them to understand what the word algorithm means and if its possible for them to understand what [pause] the concept of something running in a time, [pause] or… the concept of a computer program taking fifty-thousand years to finish, I think you can probably teach them this.

Me: I think the question was more like “Do you think its important?” Do you think it would be an important thing to try to teach this to – teach this to kids?

<JS>: Oh yeah! Yeah. No. I-I’m – That’s… If – I don’t think it would be an important thing to try and force it on somebody who wasn’t capable of learning it is all I was saying, but for those people who take an interest in any of these fields that have to do with computing, or any – anyone that’s in a field that would be – or, sorry; if they’re really young, anyone that has an interest in a field that could potentially end up being directly interfaced with computers or developers, then yeah, I think its important for them to at least know… the basics. You know, what it is, what could change if… it’s solved – that kind of thing.

Me: Okay.