Yeah, it's hard to give advice. At least as hard as to take advice. Don't over specialize, don't be too sure that you know the future. Be flexible and remember that careers and jobs are a long-term thing. Too many young people think they can optimize something and then they find they've spent a couple of years or more specializing in something that may not have been the right thing and in the process they burn out because they haven't spent enough time building up friendships and having a life outside computing. I meet a lot of sort of "junior geeks" that just think that the only thing that matters is the specialty of computing programming or AI or graphics or something like that and well it isn't and the rug might be pulled under them but for that. And if they do nothing else, well if you don't communicate your ideas you can just as well do Sudoku. You have to communicate and a lot of sort of caricature nerds forget that. They think that if they can just write the best code they can change the world but you have to be able to listen, you have to be able to communicate with your would-be users, and learn from them, and you have to be able to communicate your ideas to them. So you can't just do code you have to do something about culture and how to express ideas and I mean I never regretted the time I spent on history and on math. Math sharpens your mind, history gives you some idea of your limitations, on what's going on in the world, and so don't be too sure. Take time to have a balanced life and be ready for the opportunity. I mean a broad-based education of odd-based skill set, which is what you build up when you educate, you're patiently building a portfolio of skills, means that you can take advantage of an opportunity when it comes along. You can recognize it sometimes, we have lots of opportunities but a lot of them we either can't take advantage of or we don't notice. It was my fairly broad education, I've done standard computer science and compilers, I've done multiple languages, I think I knew two dozen at the time, and I have done machine architecture, I've done operating systems, and that skill set turned out to be useful.