My pursuit of computer science started small. Bored in my middle school math classes, I fiddled around with my graphing calculator. At first, I looked for patterns in numbers. Next, I found ways to compute square roots, construct Pascal’s Triangle, and solve counting problems. As my interest in math transferred almost completely to computer science, I started programming games, learning recursion, and exploring graphics. Seeing how far I progressed on a clunky calculator, it’s not hard to imagine how captivated I became when I finally started learning C++ from my dad and found out about programming competitions.
At the beginning of eighth grade, my brother introduced me to the intense world of competitive programming. For high school students in the U.S., this centers on the USA Computing Olympiad (USACO). Starting with the USACO training pages, competitors learn the basics of computer science. Each year, there are several monthly contests, dividing competitors into Bronze, Silver, and Gold divisions. At the end of each competition season, 16 students are invited to attend the USA Invitational Computing Olympiad (USAICO), a training camp where the top four competitors are then selected to represent the U.S. at the International Olympiad in Informatics (IOI).

I soaked up all of this information with curiosity, never expecting that I would reach further than the first few steps of this process. Based on what I saw on the website, I found the path to success to be incredibly daunting. Many of the top competitors had ranked internationally in math, others had started programming at a young age, most had completed the USACO training pages—none, it seemed, had gotten started by playing around with a calculator. Nevertheless, I had found a source of inspiration.

In the Company of Cows
My first year of competition was incredibly eye-opening. Although the first problems were conceptually easy, I still found C++ programming syntax difficult to decipher. At this point, I was also introduced to the idea of an algorithm. In theory, an algorithm is simple: command the computer to perform some operations on memory to translate input to some desired output. There is much more involved, however, with actually creating an algorithm and then implementing it in code. Learning about various famous algorithms online, I was baffled by how they worked and how I could construct them myself. Once I got started, however, the thrill of thinking through problems and refining my thought process consumed me. Exploring further, I found an inexhaustible variety of problems to practice on.

The first time I attended the USACO training camp, the summer before my freshman year, I was still fairly intimidated by the top competitors. Getting on a plane to Wisconsin, I was mostly concerned with the idea of traveling, living on my own, and meeting other students. Fortunately, at camp, competitors are divided into two groups: Holsteins and Guernseys (lovingly named after two breeds of cows). The Holsteins are put through a grueling course of competition, whereas the Guernseys are given easier contests and more lectures. At the end of camp, only Holsteins are eligible for IOI team selection.

Even as a Guernsey, I struggled to compete against other campers. On each contest, I found myself consistently unable to fully solve problems. Instead, I spent most of my time making simple optimizations, known as “hacks,” in order to eke out partial credit. What I found far more interesting were the lectures, which are essentially entire computer science courses packed into a couple of days. Although I’m sure none of the campers could claim that they completely understood...
the contents of the lectures, the key concepts stuck with me. Of particular interest were data structures, which are techniques for quickly storing, accessing, and querying data. As I trained over the next few years, the techniques taught in these lectures would reappear and steadily become easier to understand.

Interacting with the other competitors was just as rewarding, I found myself immersed in an entirely different culture. I’ll never forget the experiences we had, including staying up past midnight to play card games despite having a contest the next day (yet still doing well!). Even today, my friends and I still joke about mistakes we’ve made over the years, argue about different styles of code, and discuss the merits of various data structures.

Focus and Patience FTW
After two summers attending camp as a Guernsey, I was finally promoted to a Holstein. Doing fairly well among the Holsteins, I was invited to the IOI in Thailand, where I placed among the top 20 and received a gold medal. Although I was proud of my achievement, I was still struck with a sense of dissatisfaction. At camp, I had resorted to using numerous hacks to get easy points, and, at the IOI, I missed several key observations, resulting in suboptimal solutions. I left Thailand excited by the experience and eager to improve.

This past September at the IOI in Italy marked the culmination of all of these efforts. After four years of experience, this was the first time I could honestly say that I had familiarized myself with most of the techniques that could appear. There was, however, an issue: over time, the IOI has shifted toward using more ad-hoc problems—unpredictable problems that require intuition and logic more than technical knowledge. Many of these problems have no perfect solution, requiring creativity and flexibility to solve.

On the first day of competition, I put my well-practiced skills to use, patiently working through one problem after another. The most difficult problem of the day was an ad-hoc problem that introduced an “odometer” programming language reminiscent of Karel, which is typically taught in introductory Java classes. The hindrance, however, was that the language supplied only the most primitive programming operations. The contestants were restricted to navigating a 2D grid and moving stones around to act as counters. Without traditional programming constructs like variables, loops, and functions, the tasks became tremendously more difficult. Further, the problem was composed of five independent subtasks and graded using a partial-credit system. Strategically, I decided to leave it for last.

Fortunately, with focus and patience, I managed to solve each of the problems with a small amount of time to spare. Relieved at being able to eke out a perfect score in the last few minutes, I walked out preparing to be chastised for my last-minute submissions. Instead, I was greeted with cheers and warm smiles as the only perfect scorer of the day. Incredibly, I had managed to unseat the existing three-time champion, but it remained to be seen whether I could repeat the feat on day two.

On day two, faced with incredible pressure to perform, my mind was flooded with thoughts and concerns. When I finally managed to focus my attention on the actual problems, I found that they were less time-consuming than the previous day’s. The problem statements, however, were unusually lengthy, which exacerbated my anxiety. Further, tripped by simple mistakes in my thought process, my programming took much longer than it should have. Ultimately, after re-reading the problem statements and sorting out my code, I walked out as one of three perfect scorers on day two. The previous day and its seemingly unremarkable odometer had proven to be the deciding factor—I had won.

From there, my experiences were fairly anticlimactic, since the full rankings of the IOI are released as soon as the contest ends. Given a few days to tour Italy and its scenery, I had time to reflect on my experiences. Thinking about the contest, I realized that the problems were of comparable difficulty to those presented at the USACO training camp. Guided by the dedicated organizers and brilliant coaches of the USACO, I have reached heights that, four years ago, I could not have imagined.

Johnny Ho is a senior at Lynbrook High School in California and will be attending Harvard University in the fall. In his spare time, he enjoys playing cello, tossing a Frisbee around, and browsing Quora.