Computer science (CS) can help tackle many challenging problems effectively, that is why I want to pursue a career in this field. The Enigma Machine in the Second World War and the theorem prover in mathematics are both great examples of the ability of CS. To make full use of its power, I would love to learn more about this subject.

To dive deeper into CS, I attended a summer camp where I learned how to use algorithms to solve Sudoku. I had never coded up a solution to any problem before, but after the camp, I could write a program that solved Sudoku within a reasonable time. The game requires one digit from 1 to 9 to appear in each row, column and 3\*3 sub-grids. To solve the problem, one can always start with the brute-force approach which is simply to try out all possible numbers in each empty cell until we find a pattern that fulfills the game's requirement. After some learning, I was able to solve the problem using a search tree along with Depth-First Search (DFS) and Breadth-First Search (BFS). I took the initial 9\*9 Sudoku as the root and assigned values from 1 to 9 to the first empty cell one by one. When the first empty cell was assigned by the value 1, the root would branch into a new node. Then, I would check the correctness of the Sudoku to see if it met the rules. If incorrect, it would backtrack to the previous node, which was its parent node, and assign the integer 2 to this empty cell, and the root node would have a new child branch. It would continue to do so until a valid digit was found. All the other empty cells were found in this fashion. By doing this recursively, the leaves we could obtain would be valid Sudoku patterns. It is surprising that a seemingly simple problem like Sudoku already requires such a complex solution. One can imagine how sophisticated the solution can be to problems like weather forecasting and autonomous driving.

My passion did not end with Sudoku, hence I took up a real-world problem, cybersecurity fallacy detection later on with a friend of mine. We were trying to use the Support Vector Machine(SVM) to tackle SQL injection and XSS attacks. Concretely, we collected a list of URL requests which included normal URLs samples, the SQL injected samples and XSS samples from the internet. Next, we converted these samples to text files and vectorized these files using a dictionary which included keywords from SQL injections, XSS and other special characters. The objective of the task was to classify the URL requests into three categories, one was normal and the other two received attacks. We utilized a linear SVM to do the training and testing. Eventually, we managed to obtain a classification accuracy of around 85% on the test set. I understand that our methodology might not be perfect but it was a rather educational experience to have gone through the process of manually collecting all the URLs, setting up the modeling pipeline and analyzing the results. Dealing with these nitty and gritty details was not easy which gave me new perspectives on the differences between the theory and the real world and the potential value these applications can bring. I believe by studying CS in college, I will be better positioned to solve more challenging real-world problems.

Having witnessed first hand how computers can contribute in the problem-solving process, I know it is not easy to fully utilize CS's power so I participated in Math Competitions in both China and the UK to strengthen my math foundation, during which I also received awards such as the gold certificate in the UK Senior Mathematical Challenge 2018 and 2019. I hope my solid theory background along with the CS concepts that I will learn in college can one day enable me to solve the problems that no one else has solved to make this world a better place.