Perhaps everybody’s zeal for computers originated from playing video games, and I am no exception. I was crazy about a shooting game called “Counter-Strike” when I was in middle school. Apart from improving my skills of playing the game, I did plenty of research on how to change the game by modifying its source code, which intrigued my first taste of coding. The mysteries of programming triggered my aspiration to study Computer Science.

My passion for programming peaked in high school. As an inquisitive teenager, I was constantly seeking more challenges and began to learn Java by myself by reading a book called “Introduction to Java Programming” by Y. Daniel Liang. Later, when I was learning Matrix-Multiplication in Maths, I coded an algorithm using triple-nested loops and two-dimensional arrays. While I was debugging, I noticed that some calculations were duplicated which intrigued my consideration of the efficiency of algorithms. I enjoyed exploring high-efficiency algorithms by making a comparison with the old one. The Big O notation in ‘Introduction to Algorithms’ allows further development of my understanding of the efficiency of algorithms. The classic game Tower of Hanoi gave me the idea of recursion method. I found that the recursion method is an ideal means to solve some problem such as factorial. When I read the Strassen algorithm, I learnt that divide-and-conquer strategy could make the recursion tree smaller which reduce the time complexity of Matrix-Multiplication from n^3 to approximately n^2.8. Then I noticed that Matrix-Multiplication are massively used in graphics display to matrix translations. Although the time complexity of one Matrix-Multiplication is not reduced much, a huge reduction is made when a lot of calculations are involved. As I dived more into the ocean of algorithms, the many nuanced but powerful variance of algorithms constantly pushed me to explore further into the region.

Computer Science never failed to fascinate me with its seamless flow of logic. Last summer vacation, I made a personalised TicTacToe AI for my cousin. This AI was initially working through judging various situations and coded by complicated if statements. It is well known that a TicTacToe game must result in a draw when two players are playing optimally. To further hone my AI, I started to modify my algorithm by applying more advanced algorithms and I found the MiniMax algorithm. I learnt that the essence of the MiniMax algorithm is a search algorithm that fits for two-player zero-sum game. For games with small sizes, MiniMax algorithm can easily handle this by purposefully solving all of the possible situations to determine the best move which makes it unbeatable. In addition, the algorithm can be further optimised by combining Alpha-beta pruning which eliminates unnecessary branches to reduce the number of nodes traversed. In fact, by passing two parameters to the recursive MiniMax function and making comparisons, we could crop a node and the subtree. This efficiency optimisation process could save more than half of the original time required. The MiniMax algorithm gives me an idea of search algorithms and I started to think and solve more classical problems.

Furthermore, I have strengthened my learning skills from extra-curricular activities. In a summer program organised by the University of Cambridge in 2018, I learned Python language and basic knowledge of Machine Learning. Academically, I also received a Global Silver Award in the British Physics Olympiad, the Gold Award in the Rising Star Chemistry Challenge (top 3%) and a high distinction and credit in Australian’s Chemistry and Math Competitions.

With the knowledge I had for Computer Science, I hope to be immersed in computer science at a higher level, exploring the area that I am interested in and good at. I believe the opportunity to study in one of the most outstanding institutions will guide me in my exploration and enable me to construct new solutions to problems.