Human beings tend to plan and process their tasks in sequential order, but how computers work? Computers may solve some problems which seem to be unachievable to us. This question charmed me since the very first time I used my computer to process multiple tasks at the same time. I didn’t stop exploring the answer to that question as I grew up, and I believe it’s time now for me to learn more about Computer Science at an advanced academic level.

I attended a Computer Science summer school at the University of Cambridge in 2018. It was my first time ever doing programming. Initially, the concepts being taught such as logic control statements and loop statements seem deceivingly simple. However, when I had to combine all of these concepts together to solve problems, things got much more challenging. Among all the things that I learnt, recursion fascinated me the most as it was a concept that I had not heard before but at the same time so widely applicable. One application could be to obtain the nth term of the Fibonacci Sequence. Concretely, if n, is 1 or 2, we directly return 1, otherwise, we return the sum of the previous terms by calling the function on n-1 and n-2. Later I understood that recursion might be the answer to that question.

Nevertheless, after learning time complexity, I realise processes of recursion can be represented by a binary tree with n layers, so the maximum number of nodes is 2^n (and in fact, it is 1.618^n). Thus, the total time complexity of recursion method is O(2^n), an exponential order. It is very slow because too much time is wasted to recalculate some Fibonacci numbers which might have been calculated before. I later discovered an iterative method that stores and updates the previous two Fibonacci numbers. Thus, a decent amount of time was saved. The iteration method has time complexity O(n), which is significantly less than the exponential one. However, I was wondering whether there is an even better solution. I then discovered an O(log n) method online by using matrices, and Fibonacci sequence can, therefore, be obtained using matrix multiplication and binary division.

By using binary division, A(n) could be obtained in log n operations, and hence the total time complexity can be obtained. Through the Fibonacci sequence algorithm, I learned that repetitive operations lead to a large increase in time complexity. We can sometimes optimise algorithms by sacrificing the memory space to store useful data calculated before, as the iteration method does. This is also the idea of dynamic programming. However, there is no repetitive data to be calculated in the matrix method at all, which gives it the best performance. In some particular cases, however, it is impossible to avoid using recursion, such as the Hanoi Tower. Recursion seems to be the answer to the question I had, but the efficiency of algorithms is the key to it.

I self-studied more algorithms by reading the ‘Introduction to Algorithms’ after I got back from the summer school. I was interested in the divide and conquer topic, as the way it solves a problem is based on recursion, and therefore it totally differs from human logic. Humans prefer to use insertion sort with time complexity O(n^2). It is slower than the merge sort and QuickSort, whose time complexities are both O(log n). As I learnt more, I understood that iterations are also based on recursion, which convinced me again that the efficiency of algorithms is the key to my question. Apparently, humans rarely use divide and conquer to solve problems, because it requires too much memorisation. However, with the help of computer hardware and algorithms, a complex problem can be easily solved using recursion. Thus this principle motivates me to keep learning computer science in order to efficiently solve the existing difficult problems by computer.

In addition, I discovered the importance of creativity and learned how to express my ideas logically and explicitly by attending the ASDAN business simulation in 2017. Apart from that, I aimed to improve my math level, so I motivated myself by preparing UKMT senior mathematical challenge and Galois contest. Finally, I successfully overcome those tricky maths questions and got a gold certificate in UKMT senior mathematical challenge and Distinction in Galois contest.

I have never lost interest in studying Computer Science since I first learnt Computer Science, but instead, I cannot wait to dive deeper into this subject. I believe that I have the capability of studying at a top university in the UK.