My interests in computer science can be captured in a sentence that some quite challenging problems can be solved in a more effective way through computer science. For instance, the creation of the Enigma breaker is to do the code-breaking in World War II and some difficult theorems like the Four Colour theorem can also be proved. The fast calculation and deduction of computer are exactly beneficial to problems that can not be tackled traditionally. I expect to address those problems in a more productive way and this is the reason why I want to study this subject.

With the purpose to get insight into the field of computer science and improve my computational thinking, I attended a summer camp teaching us about how to use algorithms to solve Sudoku with trainers’ instruction. Honestly, I had no clear picture of solving the problem at the beginning, thus I first referred to how human solve the Sudoku. To be specific, look through each row, column and 3x3 sub-grid and then find the empty cell that a specific number must be filled in by using a lot of if statements. In contrast with the limited function of the original method in certain Sudoku, a hint provided by my instructor helped me come up with using the Depth First Search Algorithms (DFS) and the Breadth-First Search Algorithm (BFS). I attempted to consider the whole Sudoku as a vertex, each time assigning values from 1 to 9 to the empty cell in the Sudoku. Then, I made a function to check the correctness of the Sudoku and saved the valid ones in the stack(DFS) or the queue(BFS). If correct, it would go to the next blank and the whole Sudoku would be saved. Else, it would backtrack to the previous one. By doing this recursively, we could work out the correct answers. After testing, I surprisingly found that DFS is quicker than BFS in most cases. I thought it was the reason that DFS aims to go as far as possible whereas BFS discovers all vertices at a certain distance before moving to the further vertices and especially in the case of Sudoku, the number of possibilities and the width of the graphs were really large. In this way, DFS was more suitable for solving Sudoku. Besides, I gained a profound understanding of the time complexities of different algorithms, the logics and the connection with the relevant problems.

Unfortunately, the Sudoku problem has proved to be an NP-complete problem. Though it may validate the feasibility of a solution quickly, it still takes a long time to find the solution. Thus, to speed it up, I made an effort to avoid some unnecessary vertices through pruning. Specifically, I made a function to create a list containing 1-9 integers in it, and each time the candidate Sudoku was popped out from the stack or queue. If having appeared in the other cells in the row, column or 3x3 sub-grids, integers would be removed from the list. Therefore, testing a large number of possibilities was reduced. Just like cutting the branches of the tree, the width of the graph and the processing time fell dramatically. Thanks to this experience, I was able to conduct the projects and handle data at an early age. Another benefit is that it also trained my innovative and analytical ability by searching multiple solutions to a problem.

Apart from computer science, I am also very interested in Mathematics and Physics. I achieved a gold certificate in the UK Senior Mathematical Challenge 2018 and 2019 and the First Place Certificate for the Problem-Solving Round in the High School Team Maths Competition in 2018, facilitating me to form a systematic understanding of some mathematical concepts in computer science like the Asymptote and Big-O notations.

The potential of computer in analyzing data, solving numerous problems and some Intellectual challenges really inspires me. What I hope to pursuit is to improve the algorithms and make it more intelligent for life convenience. I cannot wait to learn more at a higher level.