18. The End is Nigh

In my end is my beginning.
Mary Queen of Scots, embroidered motto.

The main aim of this book has been to introduce some of the main data structures and algorithms in a familiar context. We use many of the data structures and algorithms without much thought, naturally choosing something suitable. At a bus stop we form a queue, faced with a pile of chairs, we add and remove from the top. If searching a shuffled pack of cards we use linear search, but given a dictionary we do something much closer to binary search.

When programming the data structures and algorithms, we are effectively just simulating their real world equivalents. Once you are familiar with the basic idea of a data structure or algorithm and can relate it to the real world, it is then much easier to understand the computer equivalent. It is also much easier to program it on a computer. The things being organised become data, and the instructions being followed must be written in a particular programming language and so must be much more precise. However, the ideas are the same.

Despite looking at a wide variety of search and sort algorithms, many more have been invented that we have not mentioned. Some are variations on the algorithms we have examined. Others are completely new. The more specialised algorithms that we have barely touched upon, such as Quicksort (one of the fastest sort algorithms devised) are rarely used in real life outside computer programs. This is because they are more complicated to follow (and similarly to program). However, they are commonly used in programs as they are so efficient. There are also many other areas where a wide range of algorithms have been devised. We have also only touched on some of the most important aspects of algorithms: how you determine their efficiency. We have noted for example that binary search is significantly faster than linear search and mergesort is faster than bubble sort. However, how much faster are they? Are there faster algorithms still? Are they always faster? How do we compare the space requirements of different algorithms? My hope is that this booklet has given you a glimpse of how interesting the subject of data structures and algorithms can be, and whetted your appetite to find out more.

Acknowledgements

I am grateful to Margaret Curzon, Maurice Curzon and Ann Blandford who provided some of the examples described, and to the students of COM2060, COM1501 and COM1000 on whom I tried out the ideas and who suggested many others. I also have had many useful conversations with Harold Thimbleby on the ideas contained here. A similar approach to teaching computing using interactive games and puzzles to that used in part here is taken by Bell et al.