Notes - Algorithms & Data Structures - Intro

- Dr Nick Hayward

A general intro to algorithms and data structures.

Contents

Intro

As we review algorithms, we may consider their usage in the context of application development.

This includes algorithms and data structures, which may work together to solve defined problems.

As such, we may initially consider an algorithm as a way to solve problems, and data structures as a storage option.

So, the data structure will store the information associated with the problem, and work in tandem with the algorithm.

A common use for data structures and algorithms includes data sorting and searching. These are basic to many structures, including stacks, queues, priority queues, bags, and so on.

We may then consider common algorithms for sorting, effective methods for organising data. For example, *quicksort*, *mergesort*, *heapsort*, and so on.

These algorithms and structures naturally help with search, including classic options such as *binary search trees*, *hash tables* &c.

They may also form part of algorithms for advanced tasks such as graph traversal and searching, shortest path algorithms, minimum spanning trees, text manipulation and processing, data compression, and so on.

As we develop custom algorithms, these fundamental structures and algorithms will form the foundations.

Why algorithms

A noticeable benefit of algorithms is their scope and application to many diverse disciplines, and their inherent abstraction.

From application development to the internet, science, social networks, video games, music, and so on, we see their use in almost every aspect of modern life and culture.

They form an invaluable part of scientific research, art, and the humanities.

A brief history of algorithms

Likewise, the history of algorithms is fascinating to consider.

The word itself, *algorithm*, has its roots in the 9th century with the mathematician *Abdullah Muhammad bin Musa al-Khwarizmi*. A mathematician, scientist, and astronomer, *al-Khwarizmi* is often noted as the father of algebra and the origin of today's word *algorithm*.

It's believed a 12th century Latin translation of a book by *al-Khwarizmi* provided a translation of his name as *Algorithmi*. There are various alternatives of this story, but the underlying origin is consistent.

However, whilst the specific word *algorithm* began with this mathematician and translation, the general concept we now associate wih an algorithm has ancient roots.

ancient roots

The origin of the use of algorithms may be traced as far back as Babylonian and Egyptian mathematics.

Whilst Babylonian and Egyptian mathematics are often considered within the same context of early mathematical usage, the Babylonian system was, in many respects, more advanced.

For example, the Babylonians were able to work with the following

- square and cube roots
- Pythagorean triples - 1200 years before Pythagoras
- knew of the existence of `pi`
- the exponential function, e - possible basic understanding
- solve some quadratics - even polynomials of degree 8
- solve linear equations
- handle measurement of circles
- ...

Suffice to say, their mathematics was not rudimentary and basic. It was concerned, to a large extent, with algebra and not geometry, an interesting contrast with the later Ancient Greeks.

The Babylonians used a sexagesimal, or `base-60`, number system inherited from the Sumerians and Akkadians.

The sexagesimal number system adopted by the Babylonians was used in conjunction with a *place* value to write numbers larger than 60. In effect, they had symbols for '1 to 59', and then repeated these symbols in additional columns to represent larger numbers.

A simple example is as follows,

column 3	column 2	column 1
2	1	9

gives us '2(60^2) + 1(60) + 9 = 7269'.

We may see a basic algorithm at work for their underlying number system.