Size Does Matter Part Three

The AQA Computing specification for A2 requires candidates to have some appreciation of the complexity of a problem defined as the growth rate of the most *efficient* algorithm which solves the problem, i.e. its big O complexity. This includes problems whose growth rates are $O(log_2\, n)$. Knowledge of logarithms is likely to be non-existent for a candidate who has not studied mathematics beyond GCSE level, However, it should be possible for candidates to become familiar with logarithms relatively quickly and then to see their use in computing. What follows is one approach that could be tried with plenty of exercises for the student to practice.

The measure of size unit the googol is 1 followed by a hundred zeros

10000000000000000........0000000000000........00000000000.......000000000

It can be written in shorthand form as 10^{100}. The shorthand form is known as the exponent or power-of-ten form, 100 is the power of ten for the googol. A decade is 1 followed by one zero, i.e. 10 or in shorthand notation 10^1 and a century is 1 followed by two zeroes, i.e. 100 written in shorthand notation 10^2. The superscripts 100, 1 and 2 in the power-of-ten representation of the googol 10^{100}, decade 10^1 and century 10^2, denote the number of zeroes present in the long-hand form 100...00, 10, 100, respectively.
The power-of-ten form is a much better system for representing large numbers compared with other systems such the Roman numeral system, not least because it makes multiplying of two numbers such as 12.5623 and 31.4902 a much easier task.

To see how this is possible in principle, the powers of ten have been laid out as shown in Figure 1 alongside their exponents (powers).

Figure 1

Power of Ten	Denary Number	Exponent
10^9	1000000000	9
10^8	100000000	8
10^7	10000000	7
10^6	1000000	6
10^5	100000	5
10^4	10000	4
10^3	1000	3
10^2	100	2
10^1	10	1
10^0	1	0
10^{-1}	1/10	-1
10^{-2}	1/100	-2

If we multiply 100 by 1000 we get 100 000 or 10^5. Now 5 = 2 + 3 but 2 and 3 are the exponents of 10^2 and 10^3 which in turn are 100 and 1000. Therefore, to multiply two numbers we first express these as powers of 10 and then add the exponents. Finally, we convert the answer as a power of ten back to normal form, i.e. 100 000.

The term log is used to remind us that we wish to work with the exponent of a number, e.g.

log 100 000 = 5 and log 100 = 2 if we are using exponents of 10, i.e. 10^5 and 10^2.

This is expressed as $\log_{10}$ 100 000 when we use exponents of 10 and $\log_2$ when we use exponents of 2, e.g. $\log_2 16 = 4$ because 16 can be expressed as 2^4.
To reverse the process we use antilog as follows:

antilog 5 = 10^5

Figure 2 shows powers of 2 alongside their exponents. The exponent still counts the number of zeroes but this time the 1 followed by zeroes represents a binary number, e.g. binary 1000000 represents the denary number 64. "Binary" means of two parts, e.g. 0 or 1, heads or tails, yes or no, true or false, the choice between two alternatives. If pennies are used with 1 represented by head and 0 represented by tail, 100000 would appear as shown in Figure 3.

Figure 3

Multiplying 100_2 by 1000_2 is achieved by adding the corresponding exponents 2 and 3 ($100_2 = 2^2$ and $1000_2 = 2^3$) to produce 5, Therefore, the answer is 2^5 or 1 followed by 5 zeroes, 100000_2. The subscript 2 means the 100000 represents a binary number with the *binary digits 0 and 1*. Binary digit is often abbreviated to *bit*. One bit is the unit of information. When there is a choice of two alternatives the choice can be coded in one bit. 0 represents one alternative and 1 the other. Figure 3 shows another way using six pennies that we can write a binary number. We say Figure 3 uses 6 bits.

Figure 4

Denary Number		Binary Number	Exponent
512	2^9	1000000000	9
256	2^8	100000000	8
128	2^7	10000000	7
64	2^6	1000000	6
32	2^5	100000	5
16	2^4	10000	4
8	2^3	1000	3
4	2^2	100	2
2	2^1	10	1
1	2^0	1	0
1/2	2^{-1}	1/10	-1
1/4	2^{-2}	1/100	-2

Figure 4 shows a balloon containing about 2.5 litres of air at a pressure of 1.25 atmospheres and a temperature of 12 °C. The Gas Laws[1] and Avogadro's constant[2], tell us that this balloon contains about 10^{23} molecules of air. Suppose we wanted to attach a label to each air molecule which would uniquely identify any air molecule to us and let's suppose we have only two alternatives to use for the label. We would need to use a binary numbering system. How many binary digits or bits would we need? Or how many pennies would we need?
There are 10^{23} molecules so we need 10^{23} labels. If we had one bit, we could label 2 molecules, the first 0 and the second 1.
With two bits, we could label 4 molecules, the first 00, the second 01, the third 10 and the fourth 11. With three bits we could label 8, four bits 16 and so on.

We can use Figure 2 to deduce that the exponent corresponds to the number of bits and the denary number the number of labels ï 2, 4, 8, 16, etc. Therefore, with *n* bits we could label 2^n molecules.

The answer is *n* bits where $10^{23} = 2^n$. How do we calculate *n*? We use logarithms as follows:

Take the logs to the base 2 of both sides:

Figure 4

$$\mathbf{\log_2 10^{23} = \log_2 2^n}$$

$\log_2 2^n$ is easy, we know the answer is the exponent *n*.

We rewrite $\log_2 10^{23}$ as $\log_2 (10 \times 10 \times 10 \,..... \times 10 \times 10 \times 10)$

To multiply we take the log, add and then antilog, i.e. reverse the process.

$$\begin{aligned}\log_2 (10 \times 10 \times 10 \,..... \times 10 \times 10 \times 10) &= \log_2 (\text{antilog}_2 (\log_2 10 + \log_2 10 +..+ \log_2 10)) \\ &= \log_2 (\text{antilog}_2 (23\log_2 10)) \\ &= \log_2 \text{antilog}_2 (23\log_2 10) \\ &= 23\log_2 10 \text{ because } \log_2 \text{antilog}_2 \text{ cancel each other}\end{aligned}$$

Therefore, $n = 23\log_2 10$

But $\log_2 10 = 3.321928$, therefore

n = 70.4 or approximately 71 bits

Proceeding in a similar way, the number of elementary particles in the universe is in power-of-ten form approximately 10^{90}. This would require about 300 bits to label each uniquely. Other large quantities also benefit from representation in powers of ten, e.g. the number of atoms making up the earth's atmosphere is about 10^{44}; the number of sand grains on Blackpool's pleasure beach is about 10^{20}. How many bits are needed to label every atom in the earth's atmosphere with a unique label[3]? How many bits are needed to label every sand grain on Blackpool's beach with a unique label[4]?

[1] Boyle's and Charles' Laws combined: PV = RT
[2] Avogadro's constant = 6.023×10^{23} molecules per Mole
[3] Earth's atmosphere = 146
[4] Blackpool's beach = 66-67