Representation of Integers
—————————————

It is a known fact that in computers binary expansion is used. More generally, integers can be represented using the so called g - adic representation.

For an integer g > 1, and a positive real number α, denote by log α (to the base g) the logarithm for base g of α. For a set M, let M^k be the set of all sequences of length k with entries from M

Theorem 1
—————

Let g be an integer g > 1. For each integer integer α, there is a uniquely determined positive integer k and a uniquely determined sequence

(a1,a2,…..ak) ∈ {0,1,…..g-1}^k

With a1≠ 0 and a = Σ a(i) g^(k-i) where I runs from 1 to k ——-[1]

In addition k = ⌊ log a ⌋ + 1and a(i) is the integral quotient of a - Σ a(j) g^(k-j) by g^(k-i) for 1<= I <= K

PROOF:

Let a be a positive integer . If a can be represented as in the above, then g ^(k-1) <= a = Σ ai g ^(k-i) = g^k - 1 < g ^k
Hence k = ⌊ log a ⌋ + 1

This proves the uniqueness of k since a is unique.
Existence and uniqueness of the sequence (a1,…..ak) by induction on k
For k = 1, set a1 = a
Then [1] is satisfied and no other choice for a1

Let k > 1. We first prove the uniqueness. If there is a representation then 0 <= a - a1g^(k-1) < g ^(k-1) and therefore 0 <= a/g^(k-1) - a1 < 1
Therefore a1 is the integral quotient of a divided by g^(k-1) and hence is uniquely determined.

Set a’ = a - a1g^(k-1) = Σ I from 2 to k ai g^(k-i) is a uniquely determined representation of a’ by the induction hypotheses.
It is also clear that a representation like [1] exists and is true.

We only need to set a1 = ⌊ a/g^(k-1) ⌋ and to take the other coefficients from the representation a’ = a - a1 g^(k-1)

The sequence (a1,a2,….ak) is called the g-adic expansion of a and the elements are called digits. Its length is k = ⌊ log a ⌋ + 1 (to the base g)

If g = 2, the sequence is called the binary expansion of a
If g = 16 then the sequence is called the hexadecimal of a
Instead of (a1 a2 …..ak) we also write a1a2….ak

We consider a few examples: 10101 = 2^4 + 2^2 + 2^0 = 21
When considering the Hexadecimal system, A,B,C, D, E, F correspond to [10,11, ….15]
So A1C is the hexadecimal expansion of 10 X 16^2 + 16 + 12 = 2588

G-adic expansion of a positive integer
Binary expansion of 105

	Q.	R.
		Where Q = quotient. R = remainder
105/2 =	52.	1
52/2 =	26.	0
26/2 =	13.	0
13/2 =	6.	1
6/2 =.	3.	0
3/2 =.	1.	1
1/2		1

Stringing the remainders from the bottom up, we have
110100

Consider the hexadecimal number n = 6EF.
The length 4 normalised binary expansions of the digits are 6 = 0110, E = 1110
F = 1111
Therefore 011011101111 is the binary expansion of n
The length of a binary expansion of a positive integer is also referred to as its binary length.
The binary length of an integer is defined to be the binary length of kits absolute value. Denoted by size(a) or size a

O and Ω Notation
—————————-

When designing a cryptographic algorithm, it is necessary to estimate how much computing time and how much storage it requires. To simplify such estimates, we introduce the O and Ω Notation.

Let k be a positive integer. X, Y ⊂ N ^ k and f : X —> R >= 0, g: Y —> R >= 0 functions
We write f = O(g) if there are positive integer B and C such that for all (n1,n2….nk) ∈ N^k with ni > B, 1 <= I <= k the following is true:
1] (n1,n2…..nk) ∈ X ∩ Y; that is f(n1,n2,….nk) and g(n1,n2,….nk) are defined
2] f(n1,n2,…nk) <= Cg(n1,n2,….nk)
This means that almost always f(n1,….nk) <= Cg(n1,…nk)

We could also write $g = \Omega(f)$. If g is constant then we write $f = O(1)$

We now consider some examples:
We have $2n^2 + n + 1 = O(n^2)$ because $2n^2 + n + 1 <= 4n^2$ for all $n >= 1$
Also $2n^2 + n + 1 = \Omega(n^2)$ because $2n^2 + n + 1 >= 2n^2$ for all $n >= 1$

If g is an integer, $g > 2$ and if $f(n)$ denotes the length of the g-adic expansion of a positive integer n , then $f(n) = O(\log(n))$ where $\log(n)$ is the natural logarithm of n
In fact, this length is $\lfloor \log n \rfloor + 1 <= \log(n) + 1 = \log(n)/\log(g) + 1$
If $n > 3$, then $\log n > 1$ and therefore $\log(n)/\log(g) + 1 < (1/\log g + 1) \log n$