

# Hash functions

---

ian.mcloughlin@gmit.ie

## Binary functions

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^*$$

- $f$  is a function that converts strings over  $\{0, 1\}$  into other strings over  $\{0, 1\}$ .
- $f$  can be viewed as a map, a subset of  $\{0, 1\}^* \times \{0, 1\}^*$  where no two first elements are equal and every element in  $\{0, 1\}^*$  is a first element.
- For example,  $(0110, 11111101) \in \{0, 1\}^* \times \{0, 1\}^*$ .

## Hash functions

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^n$$

- A (binary) hash function is a function where the output is of fixed size.
- In the example above,  $\{0, 1\}^n$  has size  $2^n$ .
- Any set of fixed size will do, but we usually use  $\{0, 1\}^n$  for some  $n \in \mathbb{N}$ .
- For example, we could use  $\{0, 1, 00, 01, 10, 11\}$  instead.

## Using Hexadecimal

The benefit of using hex is that it saves space while being easy to convert to. Every nibble becomes a hex symbol.

---

0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	A	1010	B	1011
C	1100	D	1101	E	1110	F	1111

---

### Example

---

0100	0111	0100	1101	0100	1001	0101	0100
4	7	4	D	4	9	5	4

---

Try 010010010110000101101110 in your own time.

# Common hash functions

## Input string

Galway-Mayo Institute of Technology, Dublin Road, Galway, H91 T8NW

## Outputs

---

MD5	4EC47B38AE21FD11A5E4993995097861
SHA1	67481B1FF7E145067C12B7C6C5E681CD7EFDEDD6
SHA256	BE9863B550CC931D220E7EB08B7AFE44B70ED467A5015F34ED9DECA1B84F7A2D
CRC32	FE7E4F25

---

## Try yourself

Try “Colorless green ideas sleep furiously” and the empty string.

# Properties

**Uniform:** outputs should have the same probability as each other.

**Prefect:** every input gives a different output.

**Minimal:** outputs form a coninuous range of bit strings.

**Cryptographic:** difficult to calculate the input from the output.

**Deterministic:** always the same output for a given input.

## Loop-up tables

### Exercise

Write an algorithm that counts the number of bits set in an integer.