Rust notes, algorithms and datastructures

Matt Wood
2025-12-17

Contents

1 Introduction

…

2 Bit(s)

2.1 What is a bit

1 bit is the smallest information-unit that a computer understands. It acts as an switch, that can be either **no** or **yes**.

0	1
off	on

Table 1: 1 bit

1 bit can only show two different outcomes, given by table Table 1. With 2 bits, it can show 4 different values `00, 01, 10, 11`.

Each increment in bits give the following unit:

$$2^{\text{bit}}$$
$$\text{ie. 4 bits}$$
$$2^4 = 2 \times 2 \times 2 \times 2 = 16 \text{ possible values}$$

8 bits is called 1 byte, which goes from `00000000` to `11111111`, which is equal to $2^8 = 256$ values.

2.2 How bit 1′s and 0′s relate to real world values

2.3 1 byte

Computers differentiates between signed values and unsigned values. A signed values means, it can be either a negative number and a positive number. This can be imagined as the prefix $(+, -)$ matters and is put in front of the number, meaning we know right away if it is a negative value or a positive value. So a negative value could be -20. A positive value could be $+5$.

Unsigned values strictly positive values. Meaning that regardless of the value we assume that it is a positive number, thus there is no prefix associated to that value. So we if see an unsigned value of 50 it is a positive value.

Going back to the bits, and we only think of bits the possible values as unsigned values, we get som 1 byte 256 values. If we count from 0, meaning 0 is the first value, we have 255 values left to represent, 1 byte therefore represents 0 to 255.