Bit Manipulation

A **binary number** is a number expressed in the base-2 numeral system or binary numeral system, it is a method of mathematical expression which uses only two symbols: typically "0" (zero) and "1" (one).

We say that a certain bit is **set**, if it is one, and **cleared** if it is zero.

Bitwise operators¶

- & : The bitwise AND operator compares each bit of its first operand with the corresponding bit of its second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
- | : The bitwise inclusive OR operator compares each bit of its first operand with the corresponding bit of its second operand. If one of the two bits is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
- ∧ : The bitwise exclusive OR (XOR) operator compares each bit of its first operand with the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
- ~ : The bitwise complement (NOT) operator flips each bit of a number, if a bit is set the operator will clear it, if it is cleared the operator sets it.

Examples:

```
n          = 01011000
n-1        = 01010111
--------------------
n & (n-1)  = 01010000
```

```
n          = 01011000
n-1        = 01010111
--------------------
n | (n-1)  = 01011111
```

```
n          = 01011000
n-1        = 01010111
--------------------
n ^ (n-1)  = 00001111
```

```
n          = 01011000
--------------------
~n         = 10100111
```

Set/flip/clear a bit

Using bitwise shifts and some basic bitwise operations we can easily set, flip or clear a bit. $1 \ll x$ is a number with only the x-th bit set, while $\sim (1 \ll x)$ is a number with all bits set except the x-th bit.

- $n \mid (1 \ll x)$ sets the x-th bit in the number n
- $n \wedge (1 \ll x)$ flips the x-th bit in the number n
- $n \,\&\, \sim (1 \ll x)$ clears the x-th bit in the number n