Bit Manipulation

An Important Note

You can skip this section of the course. From personal experience and others, Bit Manipulation is rarely, or almost never asked in a technical interview.

Not something you will really use on a daily basis.

However, if you are a completionist, and like being prepared for anything, no matter how unlikely, let’s continue!

Computer Science for Dummies

At end of day, computers operate just using On/Off Switches.

Computers determine On/Off by whether something is receiving Electricity (On), or not (Off).

On/Off (Receiving Electricity or not) can be represented with 1 and 0.

So how in the world can we represent numbers in a computer using just 1s and 0s?

Base 10 to Base 2

28,049 with Base 10 System

...	...	...	10,000s place	1,000s place	100s place	10s place	1s place
0	0	0	2	8	0	4	9

5 with Base 2 System

128s place	64s place	32s place	16s place	8s place	4s place	2s place	1s place
0	0	0	0	0	1	0	1

Bitwise Operators

|	OR	…?
&	AND	…?
^	XOR	…?
~	NOT	…?
<<	Left Shift	…?
>>	Right Shift	…?

| Or Bitwise Operator

sets each bit to 1 if one of two bits is 1

```
// 1 (00000001)
// 2 (00000010)
// R (00000011)


console.log(1 | 2); // 3
```

& And Bitwise Operator

sets each bit to 1 if both bits are 1

```
// 1 (00000001)
// 2 (00000010)
// R (00000000)


console.log(1 & 2); // 0
```

^ XOR Bitwise Operator

sets each bit to 1 if **only one** of two bits is 1

```
// 3 (00000011)
// 5 (00000101)
// R (00000110)


console.log(3 ^ 5); // 6
```

<< XOR Left Shift Operator

shifts left and fills empty spaces with 0

```
// 5 (00000101)
// R (00001010)


console.log(5 << 1); // 10
```

Bitwise Operators

|	OR	sets each bit to 1 if one of two bits is 1
&	AND	sets each bit to 1 if both bits are 1
^	XOR	sets each bit to 1 if **only one** of two bits is 1
~	NOT	inverts all the bits
<<	Left Shift	shifts left and fills empty spaces with 0
>>	Right Shift	shifts right