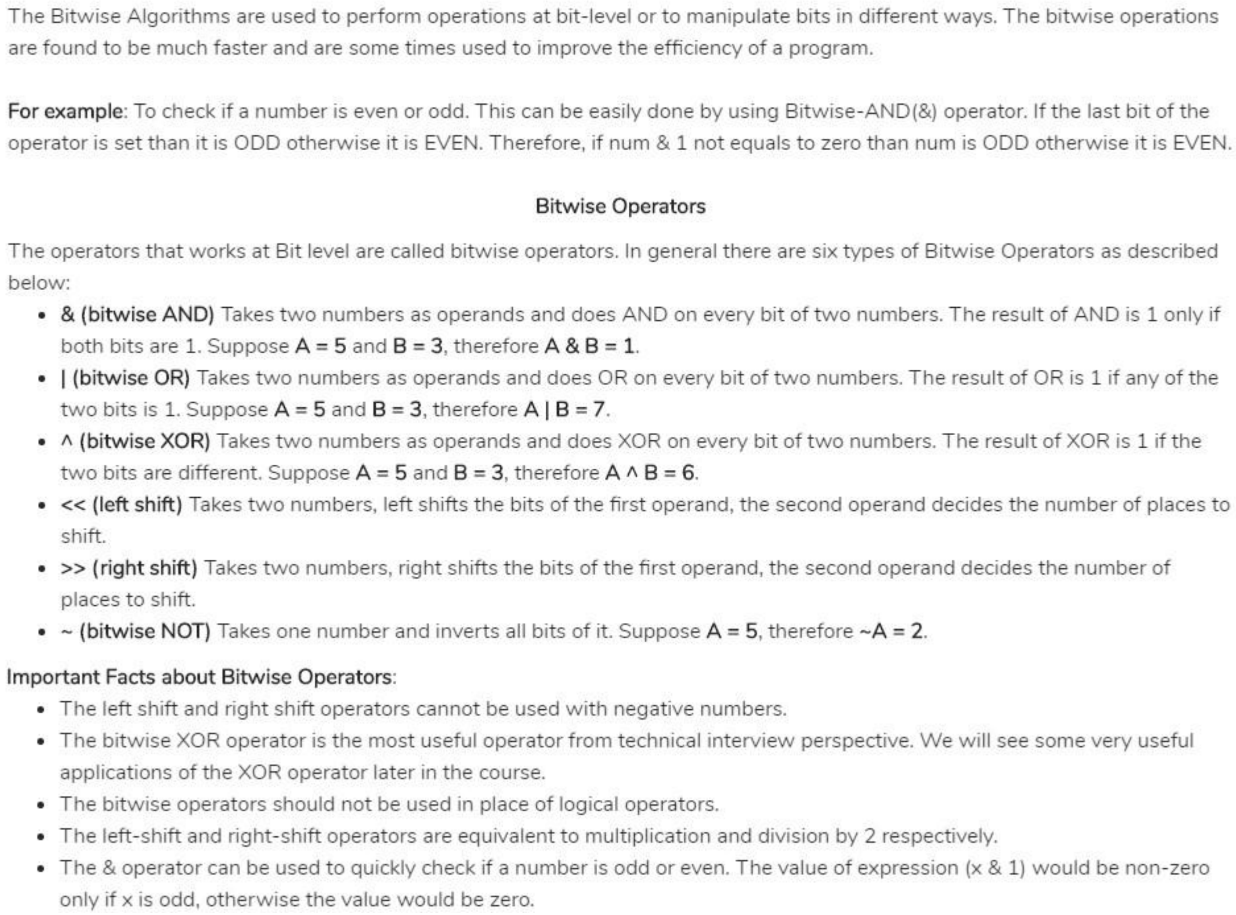
Bit Magic

Bitwise Algorithms Basics



The Bitwise Algorithms are used to perform operations at bit-level or to manipulate bits in different ways. The bitwise operations are found to be much faster and are some times used to improve the efficiency of a program.

For example: To check if a number is even or odd. This can be easily done by using Bitwise-AND(&) operator. If the last bit of the operator is set than it is ODD otherwise it is EVEN. Therefore, if num & 1 not equals to zero than num is ODD otherwise it is EVEN.

Bitwise Operators

The operators that works at Bit level are called bitwise operators. In general there are six types of Bitwise Operators as described below:

- **& (bitwise AND)** Takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1. Suppose **A = 5** and **B = 3**, therefore **A & B = 1**.
- **| (bitwise OR)** Takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 if any of the two bits is 1. Suppose **A = 5** and **B = 3**, therefore **A | B = 7**.
- **^ (bitwise XOR)** Takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different. Suppose **A = 5** and **B = 3**, therefore **A ^ B = 6**.
- **<< (left shift)** Takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.
- **>> (right shift)** Takes two numbers, right shifts the bits of the first operand, the second operand decides the number of places to shift.
- **~ (bitwise NOT)** Takes one number and inverts all bits of it. Suppose **A = 5**, therefore **~A = 2**.

Important Facts about Bitwise Operators:

- The left shift and right shift operators cannot be used with negative numbers.
- The bitwise XOR operator is the most useful operator from technical interview perspective. We will see some very useful applications of the XOR operator later in the course.
- The bitwise operators should not be used in place of logical operators.
- The left-shift and right-shift operators are equivalent to multiplication and division by 2 respectively.
- The & operator can be used to quickly check if a number is odd or even. The value of expression (x & 1) would be non-zero only if x is odd, otherwise the value would be zero.