Bitwise Operators

Bitwise operators are used to performing the manipulation of individual bits of a number which can be useful for optimizing performance in certain cases. They can be used with any integral type (char, short, int, etc.). They are used when performing update and query operations of the Binary indexed trees. For a more detailed understanding of bitwise operations.

1. Bitwise OR (|)

This operator is a binary operator, denoted by '|'. It returns bit by bit OR of input values, i.e., if either of the bits is 1, it gives 1, else it shows 0.

Example:

```
a = 5 = 0101 (In Binary)
b = 7 = 0111 (In Binary)

Bitwise OR Operation of 5 and 7
  0101
| 0111
 ________
  0111  = 7 (In decimal)
```

2. Bitwise AND (&)

This operator is a binary operator, denoted by '&.' It returns bit by bit AND of input values, i.e., if both bits are 1, it gives 1, else it shows 0.

Example:

```
a = 5 = 0101 (In Binary)
b = 7 = 0111 (In Binary)

Bitwise AND Operation of 5 and 7
  0101
& 0111
 ________
  0101  = 5 (In decimal)
```

3. Bitwise XOR (^)

This operator is a binary operator, denoted by '^.' It returns bit by bit XOR of input values, i.e., if corresponding bits are different, it gives 1, else it shows 0.

Example:

```
a = 5 = 0101 (In Binary)
b = 7 = 0111 (In Binary)

Bitwise XOR Operation of 5 and 7
  0101
^ 0111
 ________
  0010  = 2 (In decimal)
```

4. Bitwise Complement (~)

This operator is a unary operator, denoted by '~.' It returns the one's complement representation of the input value, i.e., with all bits inverted, which means it makes every 0 to 1, and every 1 to 0.

Example:

```
a = 5 = 0101 (In Binary)

Bitwise Complement Operation of 5 in java (8 bits)

~ 00000101
 ________
  11111010  = 246 (In decimal)
```

Bit-Shift Operators (Shift Operators)

Shift operators are used to shift the bits of a number left or right, thereby multiplying or dividing the number by two, respectively. They can be used when we have to multiply or divide a number by two.

Types of Shift Operators in Java:

Name of operator	Sign	Description
Signed Left Shift	<<	The left shift operator moves all bits by a given number of bits to the left.
Signed Right Shift	>>	The right shift operator moves all bits by a given number of bits to the right.
Unsigned Right Shift	>>>	It is the same as the signed right shift, but the vacant leftmost position is filled with 0 instead of the sign bit.

Entesar Abubaker