Nagarajan Senthilkumar

B.E CSE

raj.nagarajan600612@gmail.com

8778668763

Bitwise operator

Bitwise operators are used to performing the manipulation of individual bits of a number they can be used with any type like (char, int, float)

They are used when performing update and query operation of the binary tress.

There are some types of bitwise operators

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)
- Bitwise Complement (~)

This are bitwise operators and there are Bit-Shift Operator (Swift Operator)

Bit-Shift Operator (Swift Operator)

Shift operator 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 to divide a number by two

There are some types of Bit-Shift Operator

- Signed Right shift operator (>>)
- Unsigned Right shift operator (>>>)
- Signed Left shift operator(<<)
- Unsigned Left shift operator (<<<)

Now we here going to see

- Bitwise XOR (^)
- Bitwise Complement (~)

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)

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

~ 0101

_____

1010 = 10 (In decimal)

Note: Compiler will give 2's complement of that number, i.e., 2's complement of 10 will be -6.

BigInteger

About biginterger:

- The Big integer class is used to store with larger number values.
- Biginteger class in java is used for mathematical operations that require integer values larger than what primitive data types can store. A biginteger class represents immutable arbitrary-precision integers.
- byte 1byte Stores whole numbers from -128 to 127
- short 2bytes Stores whole numbers from -32,768 to 32,767

- int 4bytes Stores whole numbers from 2,147,483,648 to 2,147,483,647
- long 8bytes Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
- biginteger the only limit being the computer's memory (should be sufficient)

Program for biginteger:

```
BigInteger bi1 = new BigInteger("6378246293846238452384235456423 84");

BigInteger bi2 = new BigInteger("30397688987935472645237453792499 34");

BigInteger bigSum = bi1.add(bi2);

BigInteger bigProduct = bi1.multiply(bi2);

System.out.println("Sum : " + bigSum);

System.out.println("Product : " + bigProduct);


output

Sum: 367759352817817110976216892489 2318

Product: 193883947128790043407896524706471115960797700704819035700011960 2656
```