Bit Manipulation

Decimal (base 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Binary (base 2 : 0, 1)
9	1001
25	11001
255	11111111
$1024 = 2^{10}$	10000000000

Binary Equivalent of 155 and bit indices

7	6	5	4	3	2	1	0
1	**0**	**0**	**1**	**1**	**0**	**1**	**1**

Bitwite Operators

Operator C++	Operator Java	Meaning
&	&	AND
|	|	OR
^	^	Exclusive OR
~	~	Inversion (one's complement)
<<	<<	Left shift
>>	>>	Right shift
	>>>	Zero fill right shift (unsigned right shift)

Examples

- `10101 & 110110 = 10100`
- `10101 | 110110 = 110111`
- `byte a = 10101; cout << ^a << endl;  //prints  11101010`
- `101 << 2 = 10100  // a << b is equivalent to multiplying by a*2^b.`
- `1100 >> 2 = 11  // a >> b is equivalent to dividing by a/2^b.`

Assume if a = 60 and b = -60; now in binary format, they will be as follows −

```
a = 0000 0000 0000 0000 0000 0000 0011 1100
b = 1111 1111 1111 1111 1111 1111 1100 0100
```

In Java, negative numbers are stored as 2's complement.

```
Thus a >> 1 = 0000 0000 0000 0000 0000 0000 0001 1110
And b >> 1 = 1111 1111 1111 1111 1111 1111 1110 0010
```

Unsigned right shift operator

The unsigned right shift operator '>>' do not use the sign bit to fill the trailing positions. It always fills the trailing positions by 0s.

```
Thus a >>> 1 = 0000 0000 0000 0000 0000 0000 0001 1110
And b >>>  1 = 0111 1111 1111 1111 1111 1111 1110 0010
```

Some Useful Thechniques

Retrieving a Bit

```
//returns true if the ith bit of n is 1.
bool getBit(int n, int i)
{
	return (n & 1 << i);
}
```

Setting a Bit

```
//set the ith bit of n to 1.
int setBit(int n, int i)
{
	return (n | 1 << i);
}
```

Clearing a Bit

```
//unset (make it 0) the ith bit of n.
int clearBit(int n, int i)
{
	int mask = ~(1 << i);
	return n & mask;
}
```

Update a Bit

```
//update the ith bit of n to val (1 or 0).
int updateBit(int n, int i, int val)
{
	int mask = ~(1 << i);
	return (n & mask) | (val << i);
}
```

Obtaining the least sigificant set bit (lowest set bit)

```
int a = 20;
//declaring an 8-bit bitset
bitset<8> aa = a;          //10100
//Obtaining right-most 1 (least significant set bit)
bitset<8> bb = a & -a;     //100
// bb = x ^ ( x & (x-1)) an alternative formula.
```

Obtaining the most sigificant set bit (highest set bit)

```
//Highest set bit, zero base.
int hibit(unsigned int n) {
	//You may use a loop here
	n |= (n >> 1);
	n |= (n >> 2);
	n |= (n >> 4);
	n |= (n >> 8);
	n |= (n >> 16);
	n |= (n >> 32);

	return n - (n >> 1);
}
```

Swap two numbers a and b (XOR swapping)

```
int a = 3, b = 5;
a = a ^ b;
b = b ^ a;
a = a ^ b;
cout << a << b << endl; //5 3
```

Flipping the right most set bit

a = a & (a-1)

Counting Number of set bits (Brian Kernighan's algorithm)

n & (n-1) flips the right most set bit of the number n. Repeat the operationg until the number becomes 0.

```
1  Initialize count: = 0
2  If integer n is not zero
   (a) Do bitwise & with (n-1) and assign the value back to n
       n: = n&(n-1)
   (b) Increment count by 1
   (c) go to step 2
3  Else return count
```

XOR of three numbers

İf (a xor b == c) then a xor c == b and b xor c == a

A xor A = 0;

(A XOR B) xor C = A xor (B xor C);

C++ BitSet class and Java BitSet class

The BitSet class creates a special type of array that holds bit values. It is an array of bool but each Boolean value is not stored separately instead bitset optimizes the space such that each bool takes 1 bit space only.

BitSet contains the most common bit operations such as, couting set bits, set a bit, reset a bit, flip a bit, convert to string , convert to int etc.

Java Integer Class Bit Methods

- bitcount()
- numberOfLeadingZeroes()
- numberOfTrailingZeroes()
- highestOneBit()
- LowestOneBit()
- reverse()
- reverseBytes()

Practice Problems

1. Adding two bit strings.
2. Determine the parity of a number.
3. Toggle the m lowers bits of n.
4. Toggle the i^{th} bit of the number n.
5. Chek if a number is power of 2?
6. Generate all subsets of a set?

Useful Links

https://medium.com/@hitherejoe/bit-manipulation-b13b94e70f3b
https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/tutorial/
https://www.geeksforgeeks.org/bits-manipulation-important-tactics/