Bit Manipulation In C++

Note by Tawhid Monowar

Left shift (a<<b = a*2^b)
Right shift (a>>b = a/2^b)
For all odd numbers the last bit is 1, and for even it's 0
Odd/Even (n&1)? cout << "Odd" : cout << "Even";

Some properties of bitwise operations:

1. a|b = a⊕b + a&b
2. a⊕(a&b) = (a|b)⊕b
3. b⊕(a&b) = (a|b)⊕a
4. (a&b)⊕(a|b) = a⊕b

Addition:

1. a+b = a|b + a&b
2. a+b = a⊕b + 2(a&b)

Subtraction:

1. a-b = (a⊕(a&b))-((a|b)⊕a)
2. a-b = ((a|b)⊕b)-((a|b)⊕a)
3. a-b = (a⊕(a&b))-(b⊕(a&b))
4. a-b = ((a|b)⊕b)-(b⊕(a&b))

Get ith Bit

```
int get_ith_bit(int n, int i)
{
      int mask = (1<<i);
      return (n&mask)>0?1:0;
}
```

Clear ith Bit

```
int clear_ith_bit(int n, int i)
{
      int mask = ~(1<<i);
      n = (n&mask);
      return n;
}
```

Set ith Bit

```
int set_ith_bit(int n, int i)
{
      int mask = (1<<i);
      n=(n|mask);
      return n;
}
```

Update ith Bit

```
int update_ith_bit(int n, int i, int v)
{
      clearIthBit(n,i);
      int mask = (v<<i);
      n=(n|mask);
      return n;
}
```

Clear last i Bit

```
int clear_last_i_bit(int n, int i)
{
      int mask = (-1<<i);
      n = (n&mask);
      return n;
}
```

Clear Bits In Range

```
int clear_bits_in_range(int n, int i, int j)
{
      int a = (-1<<j+1);
      int b = (i<<i-1);
      int mask = (a|b);
      n = (n&mask);
      return n;
}
```

Update Bits In Range

```
int replace_bits_in_range(int n, int v, int i, int j)
{
      n = clear_bits_in_range(n,i,j);
      int mask = (v<<i);
      cout << (n|mask);
}
```

Check N is power of 2 or not

```
void n_is_power_of_two(int n)
{
      (n&(n-1)) ? cout << "false" : cout << "true";
}
```

Count Set Bits (Method one)

```
int count_set_bits(int n)
{
      int cont=0;
      while(n>0)
      {
          n = (n&(n-1));
          cont ++;
      }
      return cont;
}
```

Count Set Bits (Method Two)

```
int count_set_bits(int n)
{
      int cont=0;
      while(n>0)
      {
          int last_bit = (n&1);
          cont+=last_bit;
          n = n>>1;
      }
      return cont;
}
```