Bits manipulation (Important tactics)

Prerequisites : Bitwise operators in C, Bitwise Hacks for Competitive Programming, Bit Tricks for Competitive Programming

1. Compute XOR from 1 to n (direct method) :

```
// Direct XOR of all numbers from 1 to n
int computeXOR(int n)
{
    if (n % 4 == 0)
        return n;
    if (n % 4 == 1)
        return 1;
    if (n % 4 == 2)
        return n + 1;
    else
        return 0;
}
```

```
Input: 6
Output: 7
```

Refer Compute XOR from 1 to n for details.

2. We can quickly calculate the total number of combinations with numbers smaller than or equal to with a number whose sum and XOR are equal. Instead of using looping (Brute force method), we can directly find it by a mathematical trick i.e.

```
// Refer Equal Sum and XOR for details.
Answer = pow(2, count of zero bits)
```

3. How to know if a number is a power of 2?

```
//  Function to check if x is power of 2
bool isPowerOfTwo(int x)
{
     // First x in the below expression is
     // for  the case when x is 0
     return x && (!(x & (x - 1)));
}
```

Refer check if a number is power of two for details.

4. Find XOR of all subsets of a set. We can do it in O(1) time. The answer is always 0 if given set has more than one elements. For set with single element, the answer is value of single element. Refer XOR of the XOR's of all subsets for details.
5. We can quickly find number of leading, trailing zeroes and number of 1's in a binary code of an integer in C++ using GCC. It can be done by using inbuilt function i.e.

```
Number of leading zeroes: builtin_clz(x)
Number of trailing zeroes : builtin_ctz(x)
Number of 1-bits: __builtin_popcount(x)
```

Refer GCC inbuilt functions for details.

6. Convert binary code directly into an integer in C++.

```
// Conversion into Binary code//
#include <iostream>
using namespace std;

int main()
{
    auto number = 0b011;
    cout << number;
    return 0;
}
```

Run on IDE

```
Output: 3
```

7. The Quickest way to swap two numbers:

```
a ^= b;
b ^= a;
a ^= b;
```

Refer swap two numbers for details.

8. Simple approach to flip the bits of a number: It can be done by a simple way, just simply subtract the number from the value obtained when all the bits are equal to 1 .
   For example:

```
Number : Given Number
Value  : A number with all bits set in given number.
Flipped number = Value – Number.

Example :
Number = 23,
Binary form: 10111;
After flipping digits number will be: 01000;
Value: 11111 = 31;
```

9. We can find the most significant set bit in O(1) time for a fixed size integer. For example below cocde is for 32 bit integer.

```
int setBitNumber(int n)
{
    // Below steps set bits after
    // MSB (including MSB)

    // Suppose n is 273 (binary
    // is 100010001). It does following
    // 100010001 | 010001000 = 110011001
    n |= n>>1;

    // This makes sure 4 bits
    // (From MSB and including MSB)
    // are set. It does following
    // 110011001 | 001100110 = 111111111
    n |= n>>2;

    n |= n>>4;
    n |= n>>8;
    n |= n>>16;

    // Increment n by 1 so that
    // there is only one set bit
    // which is just before original
    // MSB. n now becomes 1000000000
    n = n + 1;

    // Return original MSB after shifting.
    // n now becomes 100000000
    return (n >> 1);
}
```

   Refer Find most significant set bit of a number for details.

10. We can quickly check if bits in a number are in alternate pattern (like 101010). We compute n ^ (n >> 1). If n has an alternate pattern, then n ^ (n >> 1) operation will produce a number having set bits only. '^' is a bitwise XOR operation. Refer check if a number has bits in alternate pattern for details.

This article is contributed by **Sanchit Garg 1**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Practice Tags : Bit Magic

Article Tags : Bit Magic Competitive Programming

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content. Login to Improve this Article

Recommended Posts:

Bitwise Operators in C/C++
Bitwise Hacks for Competitive Programming
Bit Tricks for Competitive Programming
Calculate XOR from 1 to n.
Equal Sum and XOR
Remove duplicates from a string in O(1) extra space
Calculate the total fine to be collected
Sum of bitwise OR of all subarrays
Value in a given range with maximum XOR
Check if a number can be expressed as 2^x + 2^y

(Login to Rate)

2.6 Average Difficulty : **2.6/5.0**
Based on **33** vote(s)

Basic Easy Medium Hard Expert

Add to TODO List
Mark as DONE

Feedback Add Notes Improve Article

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post!

A computer science portal for geeks

710-B, Advant Navis Business Park,
Sector-142, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org