

Home	Algorithms	DS	GATE	Interview Corner	Q&A	C	C++	Java	Books	Contribute	Ask a Q	About
Array	Bit Magic	C/C++	Articles	GFactS	Linked List	MCQ	Misc	Output	String	Tree	Graph	

Next higher number with same number of set bits

Given a number x, find next number with same number of 1 bits in it's binary representation.

For example, consider $x = 12$, whose binary representation is 1100 (excluding leading zeros on 32 bit machine). It contains two logic 1 bits. The next higher number with two logic 1 bits is 17 (10001_2).

Algorithm:

When we observe the binary sequence from 0 to $2^n - 1$ (n is # of bits), right most bits (least significant) vary rapidly than left most bits. The idea is to find right most string of 1's in x , and shift the pattern to right extreme, except the left most bit in the pattern. Shift the left most bit in the pattern (omitted bit) to left part of x by one position. An example makes it more clear,

```
x = 156
```

```
10
```

```
x = 10011100
```

```
(2)
```

```
10011100
```

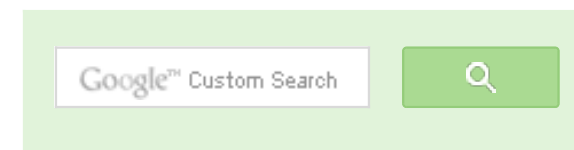
```
00011100 - right most string of 1's in x
```

```
00000011 - right shifted pattern except left most bit -----> [A]
```

```
00010000 - isolated left most bit of right most 1's pattern
```

```
00100000 - shiftleft-ed the isolated bit by one position -----> [B]
```

```
10000000 - left part of x, excluding right most 1's pattern -----> [C]
```



GeeksforGeeks



53,526 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)

```
10100000 - add B and C (OR operation) -----> [D]
10100011 - add A and D which is required number 163
```

(10)

After practicing with few examples, it easy to understand. Use the below given program for generating more sets.

Program Design:

We need to note few facts of binary numbers. The expression $x \& -x$ will isolate right most set bit in x (ensuring x will use 2's complement form for negative numbers). If we add the result to x , right most string of 1's in x will be reset, and the immediate '0' left to this pattern of 1's will be set, which is part [B] of above explanation. For example if $x = 156$, $x \& -x$ will result in 00000100, adding this result to x yields 10100000 (see part D). We left with the right shifting part of pattern of 1's (part A of above explanation).

There are different ways to achieve part A. Right shifting is essentially a division operation. What should be our divisor? Clearly, it should be multiple of 2 (avoids 0.5 error in right shifting), and it should shift the right most 1's pattern to right extreme. The expression $(x \& -x)$ will serve the purpose of divisor. An EX-OR operation between the number X and expression which is used to reset right most bits, will isolate the rightmost 1's pattern.

A Correction Factor:

Note that we are adding right most set bit to the bit pattern. The addition operation causes a shift in the bit positions. The weight of binary system is 2, one shift causes an increase by a factor of 2. Since the increased number (*rightOnesPattern* in the code) being used twice, the error propagates twice. The error needs to be corrected. A right shift by 2 positions will correct the result.

The popular name for this program is **same number of one bits**.

```
#include<iostream>

using namespace std;

typedef unsigned int uint_t;

// This function returns next higher number with same number of set bits
```



Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

// this function returns next higher number with same number of set bits
uint_t snoob(uint_t x)
{
    uint_t rightOne;
    uint_t nextHigherOneBit;
    uint_t rightOnesPattern;

    uint_t next = 0;

    if(x)
    {
        // right most set bit
        rightOne = x & -(signed)x;

        // reset the pattern and set next higher bit
        // left part of x will be here
        nextHigherOneBit = x + rightOne;

        // nextHigherOneBit is now part [D] of the above explanation.

        // isolate the pattern
        rightOnesPattern = x ^ nextHigherOneBit;

        // right adjust pattern
        rightOnesPattern = (rightOnesPattern)/rightOne;

        // correction factor
        rightOnesPattern >>= 2;

        // rightOnesPattern is now part [A] of the above explanation.

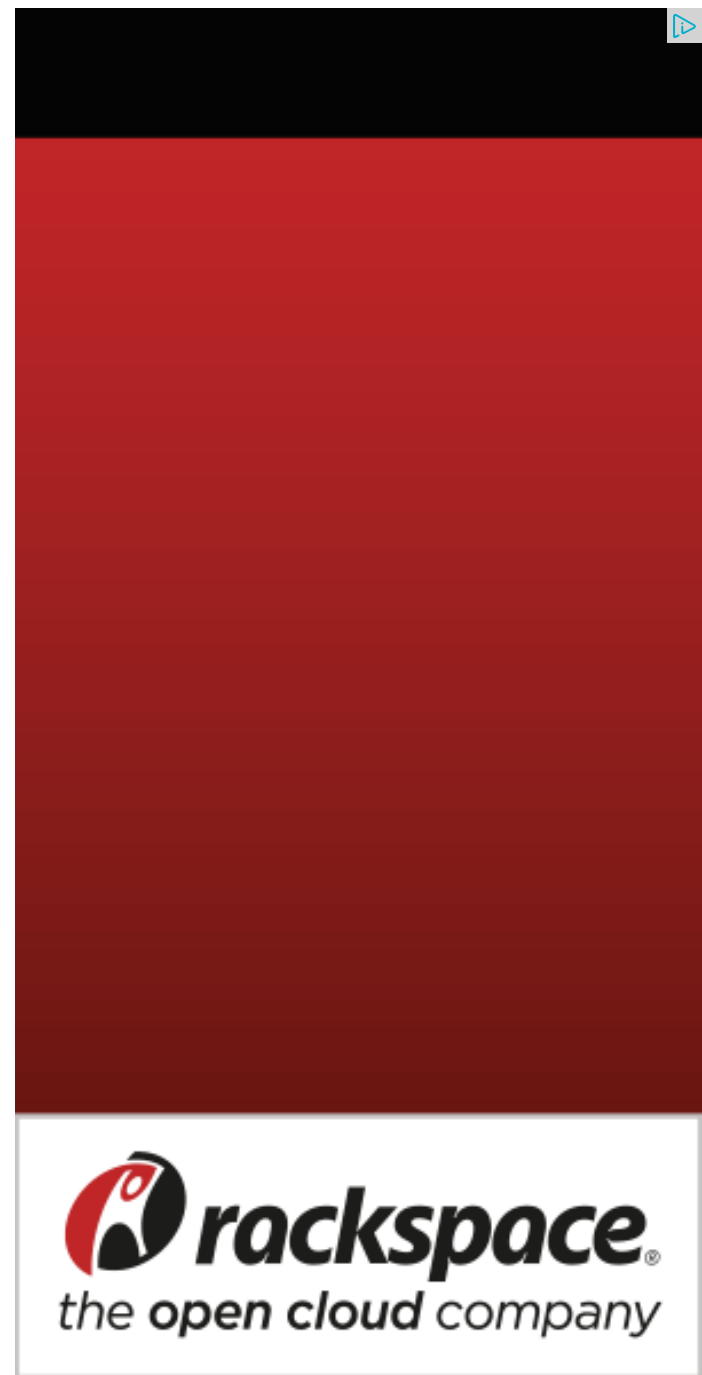
        // integrate new pattern (Add [D] and [A])
        next = nextHigherOneBit | rightOnesPattern;
    }

    return next;
}

int main()
{
    int x = 156;
    cout<<"Next higher number with same number of set bits is "<<snoob(x)

    getch();
    return 0;
}

```



}

Usage: Finding/Generating subsets.

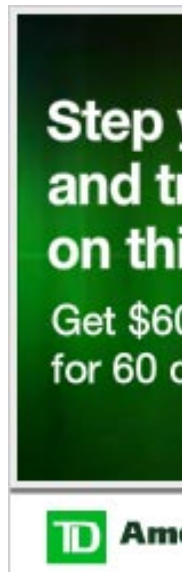
Variations:

1. Write a program to find a number immediately smaller than given, with same number of logic 1 bits? (Pretty simple)
2. How to count or generate the subsets available in the given set?

References:

1. A nice presentation [here](#).
2. [Hackers Delight](#) by Warren (An excellent and short book on various bit magic algorithms, a must for enthusiasts)
3. C A Reference Manual by Harbison and Steele (A good book on standard C, you can access code part of this post [here](#)).

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



Recent Comments

Abhi You live US or India?

[Google \(Mountain View\) interview](#) · 28 minutes ago

Aman Hi, Why arent we checking for conditions...

[Write a C program to Delete a Tree.](#) · 1 hour ago

kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 1 hour ago

Sanjay Agarwal bool

tree::Root_to_leaf_path_given_sum(tree...


[Root to leaf path sum equal to a given number](#) · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

[Count trailing zeroes in factorial of a number](#) · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...


[Find subarray with given sum](#) · 2 hours ago

AdChoices 

[► Bits Byte](#)

[► Hex Bits](#)

[► Pattern Bits](#)

AdChoices 

[► Numbers Number](#)

Related Topics:

- Check if a number is multiple of 9 using bitwise operators
- How to swap two numbers without using a temporary variable?
- Divide and Conquer | Set 4 (Karatsuba algorithm for fast multiplication)
- Find position of the only set bit
- Swap all odd and even bits
- Add two bit strings
- Write your own strcmp that ignores cases
- Binary representation of a given number



11



Tweet

0



3

Writing code in comment? Please use ideone.com and share the link here.

23 Comments

GeeksforGeeks

Sort by Newest ▼



Join the discussion...



Wellwisher • a month ago

Nice code in C#. Complexiety $O(n)$

<http://onestopinterviewprep.bl...>

^ | v • Reply • Share ›



ritu • 4 months ago

could you please explain in detail. i am still a beginner. plz

^ | v • Reply • Share ›



Omor J. Kocharee • 8 months ago

Why its too complicated?

Please check this <http://www.slideshare.net/okum> for very easv approach

► [Bits Sets](#)

► [Math Number](#)

AdChoices ▶

► [Number Counting](#)

► [Print a Number](#)

► [Even Number](#)

^ | v • Reply • Share ›



^ | v • Reply • Share ›



see more

^ | v • Reply • Share ›



First find the rightmost 1, and then search for the first 0 to the left of the rightm

first find the rightmost 1, and then search for the first 0 to the left of the rightmost 1. Change the bit on the immediate right to 0. Find the number of 1s to the left of it and add that number to the number.

For example we have 100110:

The first 1 occurs on the second last bit. and the first 0 (after the second last bit) is the bit on the right of it. Change that 0 to 1 and the bit on the right of it to 0.

100110 -> (change the 0 to 1) 101110 -> (change the bit on right of it to 0) 101010

now, count the number of 1s to the left of the 0 after 1010, we have only one 1

101010 -> 101001.

Hence the answer is 101001.

1 ^ | v • Reply • Share ›



Chen → Siddharth Rajpal • 3 months ago

This is much more straightforward ! Thanks!

^ | v • Reply • Share ›



aditya kumar • 10 months ago

first find out the pos of first rightmost '0' bit after (left of) the rightmost set bit '1'

ex : num(42) = 101010 ==> pos=3; temp= 1<>1) = 101010 + 000100 -0000

code:

```
int next_int_with_same_set_bit(int num)
```

```
t=num&(-num);
```

```
while(1)
```

```
{t<>1);
```

```
return ans;
```

```
}
```

^ | v • Reply • Share ›



Nishant Kumar · a year ago

Possible solution on GCC compilers without division operator

```
int v =12;
int t = (v | (v - 1)) + 1;
printf(" %d ",t | (1 << (__builtin_ctz(t & -t) - __builtin_ctz(v & -v)
```

^ | v · Reply · Share ›



Prateek · 2 years ago

```
/* Paste your code here (You may delete these lines if not writing cor
void nextnum(unsigned int x){
printf("%d\n", (n+(n^(n&(n-1))))|(n&(n-1)));
}
```

^ | v · Reply · Share ›



Agniswar · 3 years ago

Hi,i solved in this way..though not so efficient but pretty simple one..

Link- <http://ideone.com/NBMTG>

^ | v · Reply · Share ›



pappu · 3 years ago

```
int nextNumber(int x)
{
int u = log(x & -x);
int y = x >> u;
int z = log( ~y & -(~y));
int k = pow(2, z) - 1;
```



```
y = ((y >> 2) + 1) << (z + u);  
return y + k;  
}
```

^ | v • Reply • Share ›



AJ • 3 years ago

For immediate smaller number with same number of bits:

```
int next_lowest(int x)  
{  
    int removeones = (x + 1) & x;  
    int isolate = removeones & ~(removeones - 1);  
    int shifted = (removeones ^ isolate) | (isolate >> 1);  
    int temp = (x + 1) & ~x;  
    int factor = (shifted & ~(shifted - 1)) / (temp);  
    int toadd = (temp - 1) * factor;  
  
    return toadd|shifted;  
}
```

^ | v • Reply • Share ›



insect → AJ • 10 months ago

Your program gives wrong output for 12.

```
/* Paste your code here (You may delete these lines if not writ
```

^ | v • Reply • Share ›



insect → insect • 10 months ago

sry..its correct

^ | v • Reply • Share ›



Manish Mishra · 3 years ago

To get the immediate lower number, just use this-

```
int nextsmallest(int n)
{
    return ~nextlargest(~n);
}
```

^ | v · Reply · Share ›



Imran Amjad · 3 years ago

Hi GeeksforGeeks,

from the same logic to get the next higher number of same set bits, how i'll rev lower number? please give a brief explanation. Thanks

^ | v · Reply · Share ›



sutendra mirajkar · 3 years ago

```
#include<stdio.h>
#include<stdlib.h>

int main(int argc,char *argv[])
{
    int a,i,temp,bin,count=0,fcount=0,flag=0;

    if(argc != 2)
    {
        printf("IMPROPER EXECUTION,PLEASE TYPE IN THE NUMBER AFTER ./i
```

```
}

a=atoi(argv[1]);

int t=a;
```

[see more](#)

^ | v • Reply • Share ›



naveen kolati • 3 years ago

```
#include<stdio.h>
int naveen(int );
void main()
{
    int p=0,k,n;
    printf("enter the number");
    scanf("%d",&n);
    k = naveen(n);
    while(k!=p)
        p=naveen(++n);
    printf("\n next number is %d",n);
    getch();
}

int naveen(int n)
{
    int count=0;
    while(n!=0)
    {
        n=n&(n-1);
        count++;
    }
}
```

```
}  
    return count;  
}
```

1 ^ | v • Reply • Share ›



Himanshu • 3 years ago

Another method to find the lexicographical next permutation is given at followin

<http://graphics.stanford.edu/~...>

^ | v • Reply • Share ›



Preetam • 3 years ago

for 5 what is the next higher number?

please write few more samples

^ | v • Reply • Share ›



naveen koati → Preetam • 3 years ago

6 is the next highest number after 5

6(110),5(101)

^ | v • Reply • Share ›



Venki → Preetam • 3 years ago

@Preetam, Use the sample program for generation of sets. You can s

<http://ideone.com/W1D5E>.

^ | v • Reply • Share ›



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

Contact Us!

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team