Home » Wiki » Tutorial for bitwise operations

Tutorial for bitwise operations

Bitwise techniques

Integers are represented in base 2 in computers. To know more about how they are represented and the different data-types supported please see this page.

Bitwise operators

The bitwise operators mentioned below with examples

& (and) : Do not confuse this with the && operator. This operator is used to AND two integers. The ANDing is done bitwise, that is the i-th bit of the left operand and i-th bit of the right operand are ANDed to give the i-th bit of the result.

For e.g.

```
13 & 14 = 12
  1101
& 1110
  ------
  1100
```

| (or) : Do not confuse this with the || operator. This operator is used to OR two integers. The XORing is done similar to that of the previous two operators.

For e.g.

```
13 | 14 = 15


 1101
| 1110
  ------
  1111
```

^ (xor) : This operator is used to XOR two integers. The XORing is done similar to that of the previous two operators.

For e.g.

```
13 ^ 14 = 3


  1101
^ 1110
  ------
  0011
```

~ (not): This unary operator returns the bitwise NOT of an integer. This is not to be confused with the logical not (!) operator.

For example, (!1) returns 0 (false), but (~1) returns 0xfffffffe; the logical value of both being true.

>> (right-shift) : The operator shifts the integer n, k-bits to the right and returns that if it's used as

n >> k. New ones are appended at the end.

Right-shifting is equivalent to dividing the number by 2^k.

<< (left-shift) : The operator shifts the integer n, k-bits to the right and returns that if it's used as

n << k. New zeros are appened at the end.

Left-shifting is equivalent to multiplying the number by 2^k. Thus the expression 1<<n represents 2^n.

There is one more bit-shift operator for Java Programmers, that is the unsigned right-shift operator

>>> (unsigned right-shift) : While the signed right-shift operator (>>) shifts the bits to the right and the left most bits depends on sign extension, the unsigned right-shift operator (>>>) shifts zeros to the left most bits. Note that this operator is not in C/C++, as there are unsigned data types in C/C++ and not in Java.

How to check if the i-th bit is set in an integer

To check if a the i-th bit of a number n is set, we can see if (n & 1 << i) is non-zero. To be precise if the bit is set, then the result would be 2^i. This is because in the number 2^i there is only 1-bit and that is the i-th bit, if it's ANDing with n results in a non-zero number then the i-th bit is set in n.

How to check if an integer is a power of 2.

If x & (x – 1) is zero then the number is a power of 2. Consider this as an exercise and try to understand why it's true.

Using integers to represent sets.

Let us consider an integer having k-bits we can use this integer to represent a subset of a set of k-elements . As we know each bit is either a 0 or a 1, thus we can use this to denote whether this element belongs to this given subset or not.

Now consider a set as shown

S = { a, b, c, d }

Consider the number 0101 in binary number system (which is 5 in decimal number system).

Now we can use this number to represent this subset

X = { b, d }

This is very trivial to understand however, it is a very important technique that must be known by all programmers.
We can use it to generate all the possible subsets of a set. A quick brush up of a tutorial on permutations and combinations might help.

There are 2^k possible subsets of any given set with k-elements. What's more is that all these subsets can be described by numbers from 0 to 2^k-1 inclusive.

Thus the following pseudocode prints all the subsets of a given set (this set is known as a power set)

```
start
   A[ 0..n-1 ] //set of n-elements
   for i := 0 to $2^n$
      print “subset no. i”
      for j := 0 to n
         if j-th bit is set in i
            print A[j]
      end for
      print “\n”
   end for
end
```

Now that we have understood how to represent sets as integers. We can perform some operations on them

For e.g if set Ais represented by a and set B is represented by b

then the set a&b represents the intersection of the two sets, the set a|b represents the union of the two sets and the set a^b represents the symmetric difference of the two sets. You can think of more possible expressions such as a ^ (a & b) this means the set of elements belonging only to A and not to B.

Generating all subsets of a subset.

Consider a number 10110

The following numbers are subsets of this number

00000

00010

00100

00110

10000

10010

10100

10110

Now there's a nice and short way of generating them as shown below:

```
start
   N; //an integer
   X = N
   while true
      print X
      if( X == 0 )
         break;
      X = (X-1) & N;
   end while
end
```

Here the number X is a subset of number N.

Common bitwise tricks

x&(x-1) Returns number x with the lowest bit set off

x ^ (x & (x-1) Returns the lowest bit of a number x

x & 1<<n Returns 1<<n if the n-th bit is set in x

x | 1<<n Returns the number x with the n-th bit set

x ^ 1<<n Toggles the state of the n-th bit in the number x

Only the most Significant bit is set

Given an integer x, return a number y with only the most significant bit in x set. It is same as the largest power of 2 that is no greater than x (less than or equal to x)

Here we show, how to do this trick on a 32 bit integer. You can easily change it to work for others. Look what happens in following.

```
y = x;
y = y | ( y >>1 ) ;     y = y  | ( y >>2 );         y = y  | ( y >>4 );         y = y  | ( y >>8 );         y = y  |  ( y >>16 ) ;
```

You can see that the above right-shifting by powers of 2 and ORing at each step, sets all the bits starting from most significant bit set in x to the extreme right to 1. Now its simple to see,

y = (y + 1) >> 1;   has the required answer.

Surprisingly this is all you need to do for the problem http://www.codechef.com/problems/DCE05/ :) . See why does it work.

Karnaugh Maps and Combinational Logic Design are other deep topics which may be of interest to those reading about binary/boolean logic and bitwise operators.

Problems to practice

http://www.spoj.pl/problems/PIZZALOC

HINT: Iterate over all possible subsets and check if they are feasible, Choose the best amongst all such feasible sets.

http://www.spoj.pl/problems/CERC07B

http://www.spoj.pl/problems/DFLOOR

It doesn't matter in what order you tap the tiles, so we decide to start tapping tiles from the first row. Then iterate over all possible ways of tapping the tiles on the first row, for each subsequent row you must only tap the tiles such that the previous row is fully lighted (or switched off in the other problem).

http://www.spoj.pl/problems/SUBSUMS

Divide the given set of numbers into two sets A and B. Generate all subsets of A and find the sum of each subset and store in an array. Sort this array. Similary generate subsets of B and for each subset find it's sum and use binary search on the sorted array of subsets of A to get the required answer (Caution: Use 64-bit integers as answer can overflow a 32-bit integer)

Using integers to represent sets in a state

The following dynamic programming problems involve use of operators to represent states.

http://www.spoj.pl/problems/BABY

http://www.spoj.pl/problems/TRSTAGE

http://www.spoj.pl/problems/MMINPAID

http://www.spoj.pl/problems/M3TILE

http://www.spoj.pl/problems/HELPBOB

http://www.spoj.pl/problems/GNY07H

http://www.spoj.pl/problems/HIST2

Comments

Please login at the top to post a comment.

yellow_agony *@ 25 Dec 2009 08:51 AM*

It would be really great if some short explaination is given on representing states in dp by bits (may be using example of some problem)

Thanks in advance !

cfc *@ 25 Dec 2009 11:23 AM*

@nikhil ,the most famous problem would be a TSP prob,however here N < 16 (18 max).

Every city can be considered to be a bit.If bit is set , you visited that city ,if not set u have not visited and then u can to that city.

So,in a TSP prob you will call ur dp function with solve(start , 1 << start)where state is(current city , mask).

Then u check for base condition that whether all bits are set.If yes,u can go bck to the start,else go the cities that are unvisited.

yellow_agony *@ 29 Dec 2009 02:32 AM*

Aaha...so in large , bitmasks would be used to represent states in DP involving notion of sets . Right ?

cfc *@ 29 Dec 2009 09:45 PM*

Yes

kvenkata *@ 30 Dec 2009 11:45 AM*

w.r.t. the first DP bitmask problem (BABY) , can it be solved in better than O(n*n*2^n) ? Im getting TLE each time.....

kvenkata *@ 30 Dec 2009 12:18 PM*

apologies for the last post.... i got it.... sorry

akhilesh890 *@ 14 Aug 2010 10:11 PM*

Here is my solution for PIZZALOC in SPOJ.I am getting WA but it works fine for the sample input.

PLZ HELP!!

```
#include<iostream>
#include<stdlib.h>
#include<math.h>
using namespace std;

struct ss
{
int sx;
int sy;
int svalue;
};

struct loc
{
int x;
int y;
};
int compare(const void *p1, const void *p2);
int compare(const void *p1, const void *p2)
{

const  ss *a1 = (const ss*) p1;
const  ss *a2 = (const ss*) p2;
if(a1->svalue < a2->svalue){
return -1;
}else if(a1-> svalue == a2 -> svalue){
return 0;
}else{
return 1;
}
}
```

```
int main()
{
int k,r,m,i,j,n,count,index,people=0,kk=0,big=0,locs[25]={0};

cin >> k >> r;
cin >> m;

loc * location = new loc [m];

for ( i = 0 ; i < m ; i++)
cin >> location[i].x >> location[i].y;

cin >> n;

ss * solitaire = new  ss [n];

for ( i = 0 ; i < n ; i++ )
cin >> solitaire[i].sx >> solitaire[i].sy >> solitaire[i].svalue;
qsort( solitaire , n , sizeof(ss) , compare );

for ( i = 0 ; i <= pow(2,m) ; i++ )
{
people = 0;
count = 0;
index = 0;
for ( j = 0 ; j <  m ; j++ )
{
if((i & (1<<j)) > 0)
{
locs[index++]=j;
count++;
}
}
cout << count << endl;
if ( count == k )
{
for ( kk = 0 ; kk < n ; kk++  )
{
for ( j = 0 ; j < index ; j++ )
{
if((sqrt( (pow((location[locs[j]].x-solitaire[kk].sx),2)+(pow ((location[locs[j]].y-solitaire[kk].sy),2))))) <= r)
{
people += solitaire[kk].svalue;
break; }
}
}
if ( people >= big )
big = people;
}
cout << "people" << people << endl;
cout << "big" << big << endl;
}
cout << big << endl;
}
```

tranquility *@ 14 Nov 2010 04:28 AM*

@k.venketa-Baby can be solved in O(2^n).

prabhatsoni *@ 20 Apr 2011 01:23 AM*

Really thanks... This tutorial is just aaaaaaaaaawwwwwwwwwwwweeeeesssssssssssoooommmmmmmeeeeeeeeeeee...

javadecoder *@ 15 Aug 2011 04:46 PM*

Can somebody tell me how the problem http://www.spoj.pl/problems/GNY07H can be solved using bitmasks...,it appears to me a simple recurrence relation.

rahuldubey7474 *@ 21 Oct 2011 07:07 PM*

this is candy 1 problem of spoj i am getting wron answer please help with this #include main() { int t,i,sum,n,count; int a[100]; while(1) { sum=0,count=0; scanf("%d",&n); if(n==-1) break; else { for(i=0;i

cooltodinesh *@ 20 Sep 2012 03:24 AM*

In strage billboard problem, what if all board items are already white? shall it be damaged billboard or 0 flips.

re_hash *@ 23 May 2013 07:54 PM*

A really nice tutorial :)

satya8081 *@ 28 Jun 2013 09:43 AM*

Can anyone please explain that how is pizzaloc problem is related to bitwise concept??

mrolympia *@ 21 Jul 2013 06:14 AM*

why i am getting WA for http://www.codechef.com/problems/DCE05/ my solution http://www.codechef.com/viewsolution/2391347 . . . please help

rohitjv *@ 12 Oct 2013 01:29 AM*

awesome tutorial!

seelam_priya *@ 21 Feb 2014 05:47 PM*

it's useful for us.

nim94 *@ 16 May 2014 10:29 PM*

how to solve BABY of SPOJ?Please help!

balaji505 *@ 11 Sep 2014 12:28 PM*

nice

ritwikmishra *@ 7 Dec 2014 01:12 PM*

How to check if the i-th bit is set in an integer In this section the condition should be (n & (1 << (i-1))) Isn't it?

ritwikmishra *@ 7 Dec 2014 01:37 PM*

Also "(x&1) Returns the lowest bit of a number x" is more easy

iamprerit *@ 8 Dec 2014 10:00 PM*

really helped me..

aakash201 *@ 20 Dec 2014 11:53 PM*

i am solving spoj problem baby in o(n*2^n) then also tle each time, n does not exceeds 16 here is my solution: #include using namespace std; int n,a[100],b[100],dis[100][100],memo[16][65537]; int dfs(int i,int first) { int mini,j,x,num; if(i>n) return 0; if(memo[i-1][first]!=-1) return memo[i-1][first]; mini=INT_MAX; for(j=1;j<=n;j++) { num=1<<(j-1); if(((num)&(first))==0) { x=dis[i][j]+dfs(i+1,(first^(num))); if(x

sankar95 *@ 25 Dec 2014 08:02 PM*

@ritwikmishra, to check if the i-th bit is set in an integer, the condition should be (n & 1 << i). The least significant bit is taken as the zeroth bit. For ex: Let us consider n = 4 (Binary representation: 100), If i = 2, it means that you are referring to the 3rd bit from right. So, (n & 1 << i) = 4, which means that the bit is set.

shubhamsingh *@ 5 Jan 2015 04:48 PM*

how to solve http://www.spoj.pl/problems/HELPBOB using bitmask?

CodeChef is a non-commercial competitive programming community

About CodeChef | About Directi | CEO's Corner | C-Programming | Programming Languages | Contact Us

The time now is: 04:21:02 PM
Your Ip: 117.199.157.209

CodeChef - A Platform for Aspiring Programmers

CodeChef was created as a platform to help programmers make it big in the world of algorithms, **computer programming** and **programming contests**. At CodeChef we work hard to revive the geek in you by hosting a **programming contest** at the start of the month and another smaller programming challenge in the middle of the month. We also aim to have training sessions and discussions related to **algorithms, binary search**, technicalities like **array size** and the likes. Apart from providing a platform for **programming competitions**, CodeChef also has various algorithm tutorials and forum discussions to help those who are new to the world of **computer programming**.

Practice Section - A Place to hone your 'Computer Programming Skills'

Try your hand at one of our many practice problems and submit your solution in a language of your choice. Our **programming contest** judge accepts solutions in over 35+ programming languages. Preparing for coding contests were never this much fun! Receive points, and move up through the CodeChef ranks. Use our practice section to better prepare yourself for the multiple **programming challenges** that take place through-out the month on CodeChef.

Compete - Monthly Programming Contests and Cook-offs

Here is where you can show off your **computer programming** skills. Take part in our 10 day long monthly **coding contest** and the shorter format Cook-off **coding contest**. Put yourself up for recognition and win great prizes. Our **programming contests** have prizes worth up to Rs.20,000 and $700lots more CodeChef goodies up for grabs.

Discuss

Are you new to **computer programming**? Do you need help with algorithms? Then be a part of CodeChef's Forums and interact with all our programmers - they love helping out other programmers and sharing their ideas. Have discussions around **binary search, array size, branch-and-bound, Dijkstra's algorithm, Encryption algorithm** and more by visiting the CodeChef Forums and Wiki section.

CodeChef Community

As part of our Educational initiative, we give institutes the opportunity to associate with CodeChef in the form of Campus Chapters. Hosting **online programming competitions** is not the only feature on CodeChef. You can also host a **coding contest** for your institute on CodeChef, organize an **algorithm** event and be a guest author on our blog.

Go For Gold

The Go for Gold Initiative was launched about a year after CodeChef was incepted, to help prepare Indian students for the **ACM ICPC** World Finals competition. In the run up to the **ACM ICPC** competition, the Go for Gold initiative uses CodeChef as a platform to train students for the **ACM ICPC** competition via multiple warm up contests. As an added incentive the Go for Gold initiative is also offering over Rs.8 lacs to the Indian team that beats the 29th position at the **ACM ICPC** world finals. Find out more about the Go for Gold and the **ACM ICPC** competition here.