

Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number

Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-01/msg03888.html>

- *From:* "Ed Hook" <hook@xxxxxxxxxxxxx>
 - *Date:* 24 Jan 2006 10:28:34 -0800
-

Ujjaval wrote:

```
> Yes I need the numbers with at the most 4 1s or less than that. so i
> also need numbers with 3 1s, 2 1s and 1 1s.
>
>
> i didn't really get what Erik Naggum suggested. But I worked out the
> following way:
>
>
> n = 2^14
> count = 0
>
> for ( i=0;i<n;i++)
> {
> for (j=0;j<n;j=j*2)
> {
> if (i AND j) = j
> count = count + 1
> }
> if (count <=4)
> print i
> }
>
> This will print all the numbers having at most 4 1s in its equivalent
> binary number.
```

Actually, it won't --- note that you never reset 'count' to 0 ...

```
>
> What do you think of this??
```

It's probably worth using the well-known algorithm for counting the number of bits set in a word (and shortcircuiting the thing when you `_know_` that you're going to skip the current candidate). Here's a C implementation:

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

Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number

Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number

```
int main(void)
{
int n_bits;
int limit = 16384;
unsigned long word;
unsigned long cword;

for ( word=1 ; word <= limit ; ++word ) {
n_bits = 0;
cword = word;
while ( cword ) {
cword &= (cword-1);
if ( ++n_bits > 4 )
break;
}
if ( n_bits <= 4 )
printf("%ld\n",word);
}

return 0;
}
```

• **References:**

- ◆ **algorithm to identify number from 0–16384 having at most 4 1s in its binary number**
◇ From: Ujjaval
- ◆ **Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number**
◇ From: Jn
- ◆ **Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number**
◇ From: Ujjaval

- Prev by Date: **Re: uniform convergence?**
- Next by Date: **Re: figuring out a checksum method**
- Previous by thread: **Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number**
- Next by thread: **Re: algorithm to identify number from 0–16384 having at most 4 1s in its binary number**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**