

Optimization Techniques | Set 1 (Modulus)

Modulus operator is costly.

The modulus operator (%) in various languages is costly operation. Ultimately every operator/operation must result in processor instructions. Some processors won't have modulus instruction at hardware level, in such case the compilers will insert stubs (predefined functions) to perform modulus. It impacts performance.

There is simple technique to extract remainder when a number is divided by another number (divisor) that is power of 2? A number that is an exact power of 2 will have only one bit set in it's binary representation. Consider the following powers of 2 and their binary representations

2 – 10

4 – 100

8 – 1000

16 - 10000

Note those zeros in red color, they contribute to remainder in division operation. We can get mask for those zeros by decrementing the divisor by 1.

Generalizing the above pattern, a number that can be written in 2^n form will have only one bit set followed by n zeros on the right side of 1. When a number (N) divided by (2^n), the bit positions corresponding to the above mentioned zeros will contribute to the remainder of division operation. An example can make it clear,

Google™ Custom Search



GeeksforGeeks



53,526 people like GeeksforGeeks.



Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

$N = 87$ (1010111 - binary form)

$N\%2 = N \& (2-1) = 1010111 \& 1 = 1 = 1$

$N\%4 = N \& (4-1) = 1010111 \& 11 = 11 = 3$

$N\%8 = N \& (8-1) = 1010111 \& 111 = 111 = 7$

$N\%16 = N \& (16-1) = 1010111 \& 1111 = 111 = 7$

$N\%32 = N \& (32-1) = 1010111 \& 11111 = 10111 = 23$

Modulus operation over exact powers of 2 is simple and faster bitwise ANDing. This is the reason, programmers usually make buffer length as powers of 2.

Note that the technique will work only for divisors that are powers of 2.

An Example:

Implementation of circular queue (ring buffer) using an array. Omitting one position in the circular buffer implementation can make it easy to distinguish between *full* and *empty* conditions. When the buffer reaches SIZE-1, it needs to wrap back to initial position. The wrap back operation can be simple AND operation if the buffer size is power of 2. If we use any other size, we would need to use modulus operation.

Note:

Per experts comments, premature optimization is an evil. The optimization techniques provided are to fine tune your code after finalizing design strategy, algorithm, data structures and implementation. We recommend to avoid them at the start of code development. Code readability is key for maintenance.

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



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!](#)

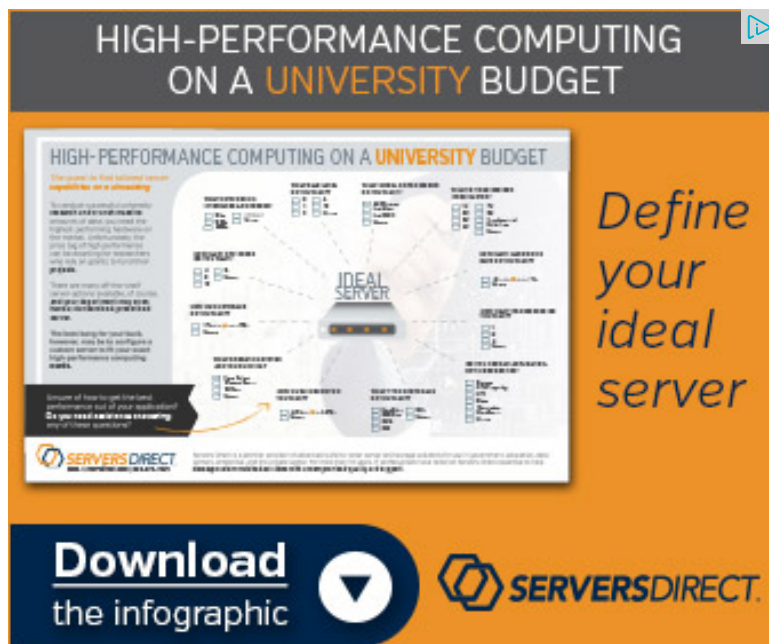
[Structure Member Alignment, Padding and](#)

[Data Packing](#)

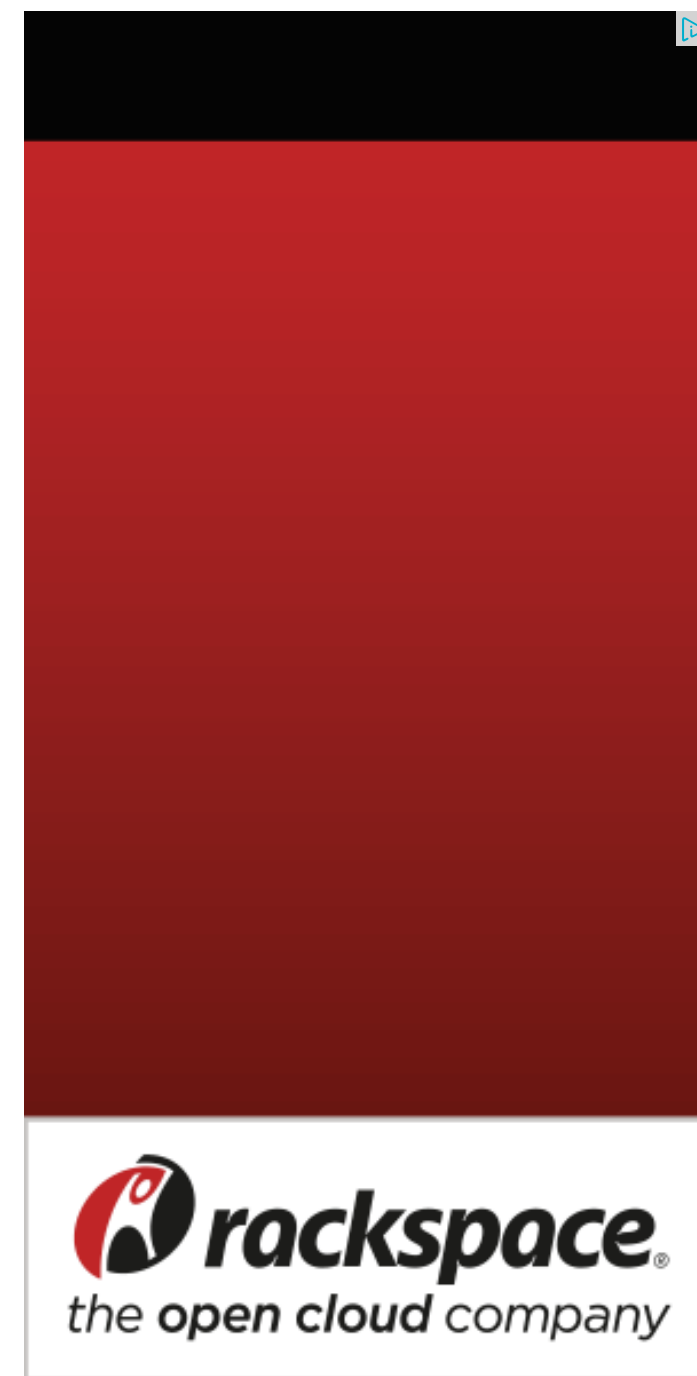
[Intersection point of two Linked Lists](#)

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

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



Sorted Linked List to Balanced BST



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



0



Tweet

0



0

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

5 Comments

GeeksforGeeks

Sort by Newest ▼



open in browser

PRO version

Are you a developer? Try out the [HTML to PDF API](https://pdfcrowd.com/html-to-pdf-api/)

pdfcrowd.com



with the algorithm...



Vibhu Tiwari · a year ago

Thanks for the code its really a good logic

^ | v · Reply · Share ›



vinay polisetti · 3 years ago

Though the logic is silly but it is worth knowing it !!

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

^ | v · Reply · Share ›



Abhirup Ghosh · 3 years ago

This solution does not work for N being -ve numbers.

^ | v · Reply · Share ›



Venki ➔ Abhirup Ghosh · 3 years ago

@Abhirup, thanks for pointing. Signed numbers will have different bit pattern. Most processors support 2's complement form of -ve numbers.

The above explanation holds for unsigned integers, and with little modification for negative numbers.

^ | v · Reply · Share ›



prog ➔ Venki · a year ago

Why don't u mention this fact in the article itself? Why force per

^ | v · Reply · Share ›

705



Subscribe

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 ▶

▶ [Java Source Code](#)

▶ [Optimization](#)

▶ [Modulus](#)

► [Modulus](#)

► [Math Techniques](#)

► [Java Array](#)

► [Java Array](#)

► [Optimization On](#)

► [And Optimization](#)