

Lonely Integer

Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

All topics

1 Bitwise XOR

2 Caching

Bitwise XOR

XOR(^) is a binary operation called exclusive OR and works as

```
1^0 = 1
0^1 = 1
0^0 = 0
1^1 = 0
```

XOR by 1 can work like a toggle switch that turns 1 to 0 or 0 to 1.

Another interesting thing to note is

```
x^0 = x
x^x = 0
```

Usage:

Problem 1: Given a number N . Flip all bits in its binary representation.

Solution 1: $N \wedge ((1 \ll 32) - 1)$ considering N is 32 bit integer.

Problem 2: Given two numbers A and B . Swap A and B without using airthmetic operator and without using third variable.

Solution 2:

$$A = A \wedge B$$
$$B = A \wedge B$$
$$A = A \wedge B$$

Related challenge for Bitwise XOR

Maximizing XOR



Success Rate: 96.83% Max Score: 30 Difficulty:

Solve Challenge

Caching

Caching or Indexing is a technique used to store counts of values which lie in a small range.

Example:

You are given a million numbers in a range, say $\in [1, 1000]$; now, if ordering is not important, a good way to store these value is to create an array of size 1000 and simply update counts of each index.

This has many benefits in simple array-based problems and string problems.

A more advanced topic is hashing, where large values of x can be hashed into small index values with a collision probability which, if below a certain value, can be used in practice.

Usage:

- Such mapping of elements from larger domain to comparatively smaller domain is useful when we need to compare elements only with $>$ or $<$ symbol. For example Coordinate Compression is useful in case of Longest Increasing Subsequence Problem and some other problems if $A_i \leq 10^9$ and $N \leq 10^5$.

Related challenge for Caching

Missing Numbers



Success Rate: 83.24% Max Score: 45 Difficulty:

Solve Challenge