Competitive Programming Reference

First, solve the problem. Then, write the code.

John Johnson

By

Sergio Gabriel Sanchez Valencia

gabrielsanv97@gmail.com

searleser97

Contents

BITs Manipulation **5**
Least Significant Set Bit 5

Ranges **5**
Data Structures 5
BIT 5
0.1 What is a Binary Indexed Tree 5
0.2 Idea 5

BITs Manipulation

Least Significant Set Bit

First thing we need to notice is that when we add 1 to a number N, what we are doing is just converting the first (right to left) 0-bit into a 1-bit and the 1-bits before get converted to 0-bits because $1 + 1 = 0$ with carry of 1 in binary, therefore we will be having a carry of 1-bit until we find a 0-bit.

Example:

$$00100111 + 1 = 00101000$$

Second thing we need to notice is very simple, lets start by denoting $\overline{N}$ as N with all it's bits inverted (1-bits change to 0-bit and viceversa), if we perform an AND operation between N and $\overline{N}$ we will get all bits in 0 as result.

Example:

$$N = 00100111$$
$$\overline{N} = 11011000$$

So, to achieve our main objective which is to extract the least significant bit (rightmost bit) we can just invert N and add 1 to it that will convert the first 0-bit to 1-bit so if we make an AND operation with N and $\overline{N}$ we get everything before the lsb as 0-bit and after the lsb we also get everything as 0-bit.

And we can write this as the 2's complement since what we did was just to invert bits and add one, which is just the exact definition of 2's complement.

C++ Code:

```
int      int
  return   & -
```

Ranges

Data Structures

BIT

0.1 What is a Binary Indexed Tree

Binary Indexed Tree (BIT) or Fenwick Tree is a data structure that can efficiently update elements and compute prefix sums in an array.

It supports query and update in $O(lg(N))$ time with $O(N)$ space complexity and it requires just a few lines of code!

0.2 Idea