Q. Solve → $T(n) = 2T(\sqrt{n}) + \log n$

assume, $n = 2^m$ → take log both side

$\log n = m$

$T(n) = T(2^m) = 2T(\sqrt{2^m}) + m$

$= T(2^m) = 2T(2^{m/2}) + m$

assum → $\underbrace{S(m)} = \underbrace{T(2^m)}$

$S(m) = 2S\left(\frac{m}{2}\right) + m \rightarrow$ direct master theorem apply

$\boxed{O(m \log m)}$

$m = \log n$

$O(\log n \log(\log n))$

Qn $T(n) = \underline{\underline{T(\sqrt{n}) + 1}}$

$$T(n) = T(m) = T(2^{m/2}) + 1$$

$$S(m) = S\left(\frac{m}{2}\right) + 1$$

$\underline{\underline{O(\log m)}}$

$\underline{\underline{O(\log \log n)}}$

Bit manipulation

observant

- AND
- OR
- XOR
- NOT
- NOR
- NAND

gate

9–12

0, 1

Just 2 mins

↳ operators

Bitwise

logical

Bitwise

operand boolean logic

value - bit

a > b and c < d

false and true

bitwise

& → and →

| → or

^ → xor

>> → right shift

division by 2

$$((x/2)/2)/2 = 2^3$$

→ ~ negation

68 → 1000100

34 → 100010

17 → 10001

8 → 1000

x >> 3

$$\boxed{0\,1\,0\,0\,1\,0\,1\,1} >> 2$$ → steps

$$\boxed{0\,0\,0\,1\,0\,0\,1\,0}$$

prepend

remaining

$<<$ → left shift → multiplication by 2

Ex

$(0100\ 1011) << 2$

$\underbrace{01}_{\text{remove}}\ \underbrace{00\ 1011}_{\text{remain}}$

$1 \times (2 \times 2 \times 2 \times 2)$

$\underbrace{001011}_{\text{remain}}\ \underbrace{00}_{\text{append}}$

log n

Q1 Given a value n, calculate 2^n

$1 << n \rightarrow 2^n$

Q) You've 2 players A and B, A plays with array 'a', B plays with array 'b'. Both wants to delete their respective arrays completely. To delete they can do the following multiple times, choose some subset numbers from the array, remove all of them. The cost of one operation is bitwise OR of the chosen no's. You've to minimize the cost for them and the one with greater value of ($\sum a[i]$ − Total cost) wins. Predict who wins if they play optimally.

Q) You've 2 players A and B, A plays with array 'a', B plays with array 'b'. Both wants to delete their respective arrays completely.

To delete they can do the following multiple times,

choose some subset numbers from the array, remove all of them. The cost of one operation is bitwise OR of the chosen no's. You've to minimize the cost for them and the one with greater value of ($\sum a[i]$ − Total cost) wins.

Predict who wins if they play optimally.

$N \le 10^5$

$a_i \le 10^5$

$b_i \le 10^5$

B 3

Ex

a → 1, 2, 1 → 1+2 → 3   4−3 ⇒ 1

b → 1, 2, 3 → 3   6−3 ⇒ 3

2 | 2 → 2

cost minimize

$[a_1, a_2 \ldots\ldots a_n]$

unique → $x_1 \to y_1$

$x_2 \to y_2$

$x_3 \to y_3$

$x_i \to y_i$

$x_i \to$ unique

$x_1 \leftarrow [x_1, x_1, x_1 \ldots x_1]$

$x_2 \leftarrow [x_2 \ldots\ldots x_2]$

[total array]

$x_1 x_1 \ldots x_1, x_2 x_2 \ldots x_2$

$x_3 x_3 \ldots\ldots x_3,$

$(x_1 + x_2 + x_3 \ldots x_n)$

1, 2, 1

1, 1, 1 ... 1, 2, 2 ... 2, ...

unordered set → { }

x | x → x

cost → (1 | 2) + (1)

→ (1 | 1) + (2)

0 !

1 ← {1, 1 ... 1} sorted

{1, 2, 2 ... 2} → 2

→ cost → (1 | 1 | 1 ... | 1) + (2 | 2 | ... | 2) + (3 | 3 .... 3) ......

{3, 3, ... 3} → 3

1 + 2 + 3 ......

Q: Check if the K^{th} bit is set or not?

$4\ 3\ 2\ 1\ 0$

$n = 01001011$ $\qquad K = 4$

$((n >> K) \& 1) \to 0 \to$ not set

$\to 1 \to$ set

$1011 \& 1$

0001

$n \& (1 << K) \to 0 \to$ not set

else $\to$ set

Q → set the i^{th} bit

$$n = n \mid (1 << k)$$

Q → toggle the k^{th} bit

$$n = n \wedge (1 << k)$$

xor → toggle

1 → 0

0 → 1

Q. Given a no. n, count it's **set bits.**

$n \rightarrow (\text{- - - - - - - - - -})$

↳ on ?? → count++

$O(\log n)$

Brian Kernighan algo

The relation of n and $n-1$

in $(n-1)$ we have almost all the bits of n same except the rightmost 1 in n, and all the bits to the right of rightmost 1.

S set bit

$n-1$ → rightmost bit and every thing to the right is flipped.

→ $n \& n-1$ S run $O(\#\text{ of set bit})$

1 set bit neutralise → unset

$$
\begin{array}{lll}
134 \rightarrow & \overbrace{10000}\overbrace{110} & \rightarrow \text{cancel out} \\
\& \; 133 \rightarrow & 10000101 & \\
\hline
 & 10000100 &
\end{array}
$$