08 May 2024 21:23

Bit Manipulation

& → and

A	B	A & B
1	1	1
1	0	0
0	1	0
0	0	0

| → Or

A	B	A | B
1	1	1
1	0	1
0	1	1
0	0	0

^ → XOR

A	B	A ^ B
1	1	0
1	0	1
0	1	1
0	0	0

<< → left shif operator

3 << 1 → 0110 → 2^2+2^1 → 4+2 → 6

$2^3\ 2^2\ 2^1\ 2^0$

↓

0011

0

Q Find out how many bits of an integer (32 bit)

Q. Find out how many bits of an integer (32 bits) are set (1)?

(28)

```
X
count = 0;
for (int i=0; i<32; i++) {
    int num = (1<<i);
    if ((X & num) > 0) {
        count++;
    }
}
return count;
```

13 → (1101)$_2$ → 3

	i	num	count
1101 & 0001	0	0001	& 1
1101 & 0010	1	0010	1
1101 & 0100	2	0100	2
1101 & 1000	3	1000	3

* Assume integers are 4 bits.

Single number

[1, 2, 2, 3, 1] → 3

① $A \wedge A \Rightarrow 0$

② $0 \wedge A \Rightarrow A$

③ $(A \wedge B) \wedge C = (A \wedge C) \wedge B$

```
num = 0
for (int i=0; i<A.size(); i++) {
    num = num ^ A[i];
}
```

0 ^ 1 ^ 2 ^ 2 ^ 3 ^ 1

1 ^ 2 ^ 2 ^ 3 ^ 1

3

return num;

$O(n)$ & $O(1)$

$1 \wedge 1 \wedge 2 \wedge 2 \wedge 3$

$0 \wedge 0 \wedge 3$

3

Single number II

[1, 2, 4, 3, 3, 2, 2, 3, 1, 1] → 4

```
for(int i=0; i<A.size(); i++) {   // A[i]
    int count=0;
    for(int j=0; j<A.size(); j++) {   // occurences of A[i]
        if(A[i] == A[j]) {
            count++;
```

[3, 3, 3, 4] (i, j)

1

3

3

```
if(count == 1) {
    return A[i];
```

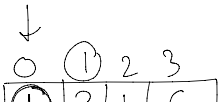
3

3

$O(n^2)$ Time $O(1)$ Space

* Assume: Integers are 4 bits long

A → [10, 10, 8, 10, 8, 5, 8] → 5



$x\%3 != 0$

num	i	1<<i	count	j	A[j] & num
		0010			

3 2 1 0

1010 → 10

1000 → 8

→ 5

num	i	1<<i	count	j	A[j] & num
0001	1	0010	0	1	0

1000 → 8
0101 → 5

→ 0101

num = (1<<i) | num;

```
  ↓
 1000
 0010
 ----
 1010
```

```
int arr[32] = {0};
for(int i=0; i<32; i++) {
    int num = (1<<i), count=0;
    for(int j=0; j<A.size(); j++) {
        if((A[j] & num)>0) {
            count++;
        }
    }
    arr[i] = count;
}
int ans=0;
for(int i=0; i<32; i++) {
    if(arr[i]%3!=0) {
```

(A[j] is set at ith location)

Time: O(32n) => O(n)

Space: O(32) => O(1)

```
if(arr[i]%3!=0){
    ans = (1<<i) | ans;
}
```

3

3

return ans;

Sum of pairwise Hamming distance

Hamming Distance : number of bits when corresponding bits are different.

10 → 7

1 1 1 1
1010 → 10
0111 → 7

+1 +1 +1 ⇒ (3)

[10, 8, 5, 7]

10,10 (0)	8,10 (1)	5,10 (4)	7,10 (3)
10,8 (1)	8,8 (0)	5,8 (3)	7,8 (4)
10,5 (4)	8,5 (3)	5,5 (0)	7,5 (1)
10,7 (3)	8,7 (4)	5,7 (1)	7,7 (0)

8 + 8 + 8 + 8 ⇒ 32

	3 2 1 0
10 →	1 0 1 0
8 →	1 0 0 0
5 →	0 1 0 1
7 →	0 1 1 1

10,8 5,7

10,5 5,10

10,7 7,10

8,5 8,7

7,8 5,8

0	1	2	3
(2)	2	2	2

(8) + 8 + 8 + 8 ⇒ 32

4 - 2 ⇒ 2

2 × 2 × 2 = (8)

2 , 2

a, b c, d

(4)

(8)

ca , cb , ac , bc
da , db , ad , bd

[10, 8, 5]

16

4 4 4
4 1 1 1
0 1 2 3
1 1 1 2

1, 2 2, 1

5 10, 8

5, 10 10, 5
5, 8 8, 5

1 1 1 1
10 → 1 0 1 0
8 → 1 0 0 0
5 → 0 1 0 1

10, 8 5

(10, 5) (5, 10)
(8, 5) (5, 8)

1 × 2 × 2 = 4

Pairs with given XOR

A = [5, 4, 10, 15, 7, 6]

10 ^ 15

B = 5

1

```
count = 0;
for (int i = 0; i < A.size(); i++) {
    for (int j = i+1; j < A.size(); j++) {
        if (A[i] ^ A[j] == B) {
```

```
if( A[i] ^ A[j] - 0 )
    count++;
}
}
}
return count;
```

T: $O(n^2)$ S: $O(1)$

if: $A \wedge B = C \rightarrow A \wedge B \wedge C = C \wedge C \rightarrow A \wedge B \wedge C \wedge B = C \wedge C \wedge B$

$A \wedge C = B$

then: $A \wedge C = B$

$A \wedge B \wedge (B \wedge C) = C \wedge (B \wedge C)$

$A \wedge C = B$

$A = [5, 4, 10, 15, 7, 6]$

$B = 5$

$5 \wedge 5 \rightarrow 0$

$4 \wedge 5 \rightarrow 0100 \wedge 0101 \Rightarrow 1$

$10 \wedge 5 \rightarrow 15$ $1010 \wedge 0101 \rightarrow 1111 \rightarrow 15$

$15 \wedge 5 \rightarrow 10$

$a \wedge b = c$

$a \wedge c = b$

map $\rightarrow [5, 4, 10, 15, 7, 6]$

1) Create a map of integers. Insert all numbers into the map.

2) count = 0.

3) Iterate through array A (i):

   a) if (A[i] ^ B is in the map):

      count++.

4) Return count / 2;

insertion : O(1)

searching : O(1)