Bit manipulation 2

7:05AM IST
6:35PM PST

- single element ✓
- single element 2 ✓
- single element 3
- Max "AND" Pair

recap:

Given an array of integers, where every element appears twise except for one element which appears Onces, find that unique element:

ex a[]: {6, 9, 6, 10, 9} ans = 10

ex a[]: {2, 3, 5, 6, 3, 6, 2} ans = 5

```
int findUnique(int a[]){
    int n = a.Len
    ans = 0
    for(i=0; i<n; i++){
n       ans ^= a[i]      ans = ans ^ a[i]
    }
    ret ans
}
```

TC: O(n)
SC: O(1)

P1 Given an array of integers(+), where every element appears twise except for two elements which apear Once, find the two unique elements: a[i] <= 10^9

ex a: {2, 3, 2, 5, 3, 6, 7, 6} {5, 7}

ex a: {1, 2, 3, 1} {2, 3}

idea 1
Brute force
TC & SC
TC: $O(n^2)$
SC: O(1)

```
for i = 0 → n-1
  for j = 0 → n-1      i ≠ j
    find dup (i, j)
```

2: 0010
3: 0011
5: 0101

6: 0110
7: 0111
1: 0001

0010 = 2
3 2 1 0
↳ b

idea 2

ex a: {2, 3, 2, 5, 3, 6, 7, 6, 1, 1}

a? b? XOR

(5^7) = 2

9^11=2

4^6 = 0100
0110
0010 = 2

XOR a[i]

① a? ≠ b? ✓

② a? ^ b? ≠ 0 ✓ c

c: 00...1...100
2 1 0
↳ b

③ a? ✓

b? 0:...0...
1:...1...
... 2 1 0

How to find exactly which bit?

groups? extra mem?

XOR1 ← groupA: {a?, 5, 1, 1}

XOR2 ← groupB: {b?, 2, 3, 2, 3, 6, 7, 6}

```
Pair<int,int> GetTwoUnique(int a[]){
    n=a.Len;  XOR=0;                          → c
    for(i=0;i<n;i++){ // a^b=XOR
n |     XOR = XOR^a[i];                        } XOR all
    }
    b=-1
    for(i=0;i<32;i++){
        if((XOR>>i)&1==1){
            b=i;
            break;
        }
    }
    XOR1=0   XOR2=0
    for(i=0;i<n;i++){
        if( a[i]>>b)&1==1)
            XOR2 = XOR2^a[i]
        else
            XOR1 = XOR1^a[i]
    }
    ret pair (XOR1, XOR2)
}
```

Quiz

TC: O(n)

SC: O(1)

P2 Given an array of integers, were every element appears trice (3 times) except for one element which appears onces, find that unique element:

$0 <= a[i] <= 10^9$

ex A: {2, 3, 6, 2, 2, 3, 3} ans=6

ex A: {5, 7, 5, 9, 7, 11, 11, 7, 5, 11} ans=9

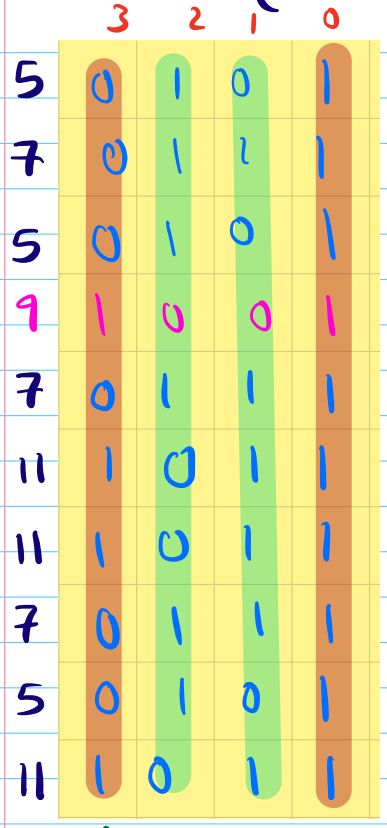
idea1

```
for i = 0 -> n-1
    for j = 0 -> n-1        i != j
        Count how many times a[i] repeated
                                (a[i] == a[j])
```

SC: O(1)

TC: O(n^2)

{5,7,5,9,7,11,11,7,5,11}



~~3x+2~~

%3

number of repeated ← R

P

number of single outlier except

GCD(R, P) = 1

add all bit in col i ← A_i % R = r

optional

Quiz
TC: O(n)
SC: O(1)

```
int findUnique3(int a[]){
    ans=0
    for(i=0;i<32;i++){          // size of var in bit length
        count=0
        for(j=0;j<n;j++){
            if( checkBit(A[j],i)) count++
        }
        if(count%3==1){
            ans = ans | (1<<i)      // setBit(ans, i)
        }
    }
    ret ans
}
```

P3 Given an array of integers, find max value of

max (a[i] & a[j]) , $i \neq j$, $1 \le a[i] \le 10^9$

ex a = { 16, 9, 11, 10 }

16 & 9 = 0 9 & 11 = 9 11 & 10 = 10

16 & 11 = 0 9 & 10 = 8

16 & 10 = 0

a & b = b & a

16 = 10000
9 = 01001
11 = 01011
10 = 01010

max(a[i], a[j]) = 10 → ans

idea 1

brute force

TC: $O(n^2)$

SC: $O(1)$

```
for i = 0 → n-1
    for j = i+1 → n-1
        ans = Max (ans, a[i] & a[j])
```

idea 2

① 8 > 7

MSB ← 1000
0111

② 8 & 11
7

1 ← 1000
0111
0 ← 1011

③ 7 & 15
6

0011
0011

0111

ex a: { 26, 13, 23, 28, 27, 7, 25 }

	4	3	2	1	0
26	1	1	0	1	0
13	0	1	1	0	1
23	1	0	1	1	1
28	1	1	1	0	0
27	1	1	0	1	1
7	0	0	1	1	1
25	1	1	0	0	1
	1	1	0	1	0 → ans

max (a_i & a_j) = ans

1 0 1 1 0

O(n)

TC: O(30n)

SC: O(1)

```
int maxAnd(int[] a){
    n = a.Len
    ans = 0
    for(b = 30; b>=0; b--){
        count = 0
        for(i=0; i<n; i++){
            count += (a[i]>>b)&1      // 0 / 1
        }
        if(count>=2){
            ans = ans | (1<<b)
            for(i=0; i<n; i++){
                if(a[i]>>b & 1 == 0){
                    a[i]=0            // → makes gray
                }
            }
        }
    }
}
```

30

check bit?

N & (1<<i) != 0