



Today's agenda

- ↳ Single element ↳
- ↳ Divide integers ↳
- ↳ man AND pairs ↳



AlgoPrep

leetcode 260



Single Number 3 ~{facebook, google, uber}

Given $arr[n]$, every element repeats twice except 2 elements. find the 2 unique elements.

↳ each occ.
1 time

$$\text{Ex: } arr[6] = \{ \underset{0}{3}, \underset{1}{6}, \underset{2}{4}, \underset{3}{4}, \underset{4}{3}, \underset{5}{8} \} \rightarrow \{6, 8\}$$

$$arr[4] = \{ \underset{0}{4}, \underset{1}{9}, \underset{2}{9}, \underset{3}{8} \} \rightarrow \{4, 8\}$$

1/idea!

↳ use Hashmap and do counting of all the elements.

T.C: $O(n)$

S.C: $O(n)$

1/idea 2

$$arr[6] = \{ \underset{0}{3}, \underset{1}{6}, \underset{2}{4}, \underset{3}{4}, \underset{4}{3}, \underset{5}{8} \}$$

$$3^6 \wedge 4^4 \wedge 3^8 \rightsquigarrow \cancel{3^8} \wedge \cancel{4^4} \wedge 4^1 \wedge 6^1 \wedge 8^1 = 6^8 = 14$$

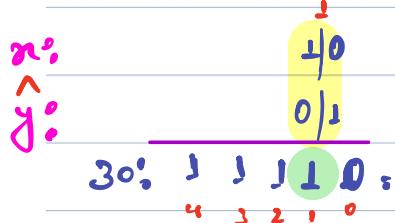
↓
14

$$14: \quad 001110 \rightarrow \begin{array}{r} 011000 \\ 000100 \\ \hline 01110 \end{array} \Rightarrow 14$$

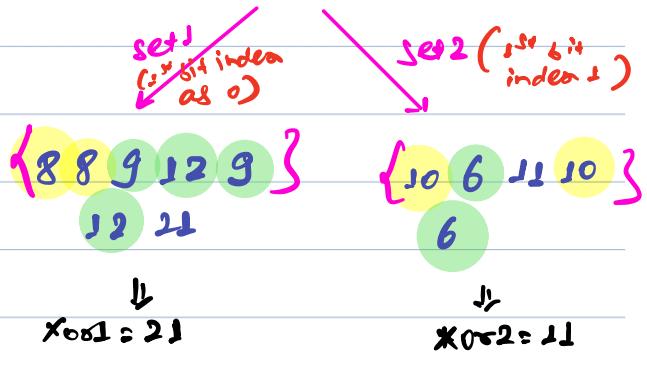
$\text{arr}[12] = \begin{matrix} 10 & 8 & 8 & 9 & 12 & 9 & 6 & 11 & 10 & 6 & 12 & 21 \\ 1000 & 1000 & 1001 & 1001 & 1001 & 1001 & 0110 & 0110 & 0110 & 0110 & 1100 & 1010 \end{matrix}$



$$\text{ans: } 11^1 21 = 20$$



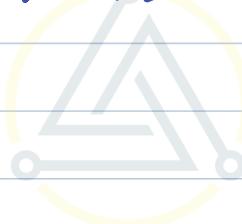
x^4 bit index (this is 1)



$$0^4 1 = 1$$

$$0^1 0 = 0$$

$$1^4 1 =$$



AlgoPrep



//Pseudo Code

```
int SingleNumber3 (int arr[N]) {
```

//Step1

```
int v = 0;
```

```
for (int i=0; i<N; i++) {
```

```
v = v ^ arr[i];
```

}

v = 20

T.C: O(N)

S.C: O(1)

//Step2

```
int idn = 0;
```

```
for (int i=0; i<32; i++) {
```

```
if (checkBit(v, i) == true) {
```

```
idn = i; break;
```

}

//Step3

```
int set1 = 0; int set2 = 0;
```

```
for (int i=0; i<N; i++) {
```

```
int num = arr[i];
```

```
if (checkBit(num, idn) == true) {
```

```
set2 = set2 ^ num;
```

```
else { set1 = set1 ^ num; }
```

3

Print (set1);

Print (set2);

3



118. Subgame Code

1/1 Step 1

$\text{arr}[12] = [10, 8, 8, 9, 12, 9, 6, 11, 10, 6, 12, 21]$

```
int v=0;
for(int i=0; i<n; i++) {
    v = v ^ arr[i];
}
```

$$\begin{aligned} v &= 10 \wedge 8 \wedge 8 \wedge 9 \wedge 12 \wedge 9 \wedge 6 \wedge 11 \wedge 10 \wedge 6 \wedge 12 \wedge 21 \\ &= 11^{\wedge} 21 = 30 \\ v &= 30 \Rightarrow 11 \ 11^{\circ} \end{aligned}$$

$idm=0;$

1/1 Step 2

```
int idm=0;
for(int i=0; i<32; i++) {
    if (checkBit(v, i) == true) {
        idm = i;
        break;
}
```

$i=0 \rightarrow \text{checkBit}(30, 0) \rightarrow \text{false}$

$i=1 \rightarrow \text{checkBit}(30, 1) \rightarrow \text{true}$

$idm=1$

$$\begin{aligned} Set1 &= 0^{\wedge} 8^{\wedge} 8^{\wedge} 9^{\wedge} 12^{\wedge} 8^{\wedge} 12^{\wedge} 21 \Sigma 21 \\ Set2 &= 0^{\wedge} 10^{\wedge} 6^{\wedge} 11^{\wedge} 10^{\wedge} 6 = 11 \end{aligned}$$

1/1 Step 3

```
int Set1=0; int Set2=0;
for(int i=0; i<n; i++) {
    int num = arr[i];
    if (checkBit(num, idm) == true) {
        Set2 = Set2 ^ num;
    } else {
        Set1 = Set1 ^ num;
    }
}
```

3

$\text{arr}[12] = [10, 8, 8, 9, 12, 9, 6, 11, 10, 6, 12, 21]$

630

Man And Pair \rightarrow { Amazon }



Q) Given $arr[n]$, find man $arr[i]$ & $arr[j]$ which can be obtained?
 $i \neq j$

Ex1: $arr[3] = \{27^{\circ} 18^{\circ} 20^{\circ}\}$

$27^{\circ}: 1 \ 1 \ 0 \ 1 \ 1$

$27 \wedge 18 = 18 \leftarrow \text{ans}$

$18^{\circ}: 1 \ 0 \ 0 \ 1 \ 0$

$18 \wedge 20 = 16$

$20^{\circ}: 1 \ 0 \ 1 \ 0 \ 0$

$27 \wedge 20 = 16$

Ex2: $arr[5] = \{21^{\circ} 18^{\circ} 24^{\circ} 17^{\circ} 16^{\circ}\}$

$21^{\circ}: 1 \ 0 \ 1 \ 0 \ 1$

$21 \wedge 16 = 16$

$16^{\circ}: 1 \ 0 \ 0 \ 0 \ 0$

$21 \wedge 17 = 17 \leftarrow \text{ans}$

$17^{\circ}: 1 \ 0 \ 0 \ 0 \ 1$

II idea

\hookrightarrow Check all Pairs and get the man out of it.

T.C: $O(n^2)$

S.C: $O(1)$

II idea2

\hookrightarrow

$90000 > 89999$

43210
 $09000 < 89999$

43210
 $10010 > 01111$

\hookrightarrow if leftmost is greater, whole number will be greater.



$\rightarrow \text{arr}[6] = \{26, 13, 23, 28, 7, 25\}$

26: 1 1 0 1 0 ↗
 4 3 2 1 0 ↗
 $020 = 0$
 $021 = 0$

13: 0 1 1 0 1
 $120 = 0$
 $121 = 1$

23: 1 0 1 1 0

28: 1 1 1 0 0

$26 \& 28 = 24$

$28 \& 25 = 24$

$26 \& 25 = 24$

7: 0 0 1 1 1

25: 1 1 0 0 1

Count: 4 3 1 1 1



11P Sieve Code

```
int manAndPair (int arr[n]) {
```

```
    for (int i=3; i>=0; i--) {
```

```
        int c=0;
```

```
        for (int j=0; j<n; j++) {
```

```
            if (checkBit(arr[j], i) == true) {
```

```
                c++;
```

```
}
```

```
        if (c>=2) {
```

```
            for (int j=0; j<n; j++) {
```

```
                if (checkBit(arr[j], i) == false) {
```

```
                    arr[j] = 0;
```

```
}
```

```
3
```

```
int idn1 = -1;
```

```
int idn2 = -1;
```

```
for (int k=0; k<n; k++) {
```

```
    if (arr[k] > 0 && idn1 == -1) {
```

```
        idn1 = k;
```

```
    else if (arr[k] > 0 && idn1 != -1) {
```

```
        idn2 = k;
```

```
        break;
```

```
3
```

return arr[idn1] & arr[idn2];

3



$$\rightarrow \text{arr}[i] = \{26 \ 13 \ 23 \ 28 \ 7 \ 25\}$$

for (int i=31; i>=0, i--) {

 int c=0;

 for (int j=0; j<N; j++) {

 if (checkBit(arr[j], i) == true) {
 c++;

 }

 if (c>=2) {

 for (int j=0; j<N; j++) {

 if (checkBit(arr[j], i) == false) {
 arr[j]=0;

 }

idn1=10
idn2=13

26 0 1 1 0 1 0

13 0 0 0 0 0 0

23 0 0 0 0 0 0

28 1 1 1 0 0

7 0 0 0 0 0 0

25 1 1 0 0 0 0

c=0123

3

int idn1=-1;

int idn2=-1;

for (int k=0; k<N; k++) {

 if (arr[k]>0 && idn1 == -1) {

 idn1=k;

 } else if (arr[k]>0 && idn1 != -1) {

 idn2=k;

 }

26 & 28



Q) Divide integers

↳ Divide two integers without using multiplication, division or mod operators.

$$\text{Ex: } 30 / 7 = 4$$

$$48 / 5 = 9$$

$$5 / 2 = 2$$

1/idea



$$\begin{array}{r} 48 \\ + 48 - 5 \\ \hline 43 \end{array}$$

$$42 - 5 = 37$$

$$37 - 5 = 32$$

$$32 - 5 = 27$$

ans=9

T.C: O(ans)

S.C: O(1)

$$22 - 5 = 17$$

$$17 - 5 = 12$$

$$12 - 5 = 7$$

$$7 - 5 = 2$$



11idea2

$$a = 48 \quad b = 5$$

$$\text{ans} = 9$$

$$g = -\frac{4}{2^3} \cdot \frac{3}{0} + \frac{2}{0} + \frac{1}{0} + \frac{0}{2^0} = 9$$

$$\text{temp} = 0 + 40 + 5$$



AlgoPrep



II) Pseudo code

```
int divideIntegers (int A, int B) {
```

```
    long a = A;
```

```
    long b = B;
```

```
    int sign = 1;
```

```
    if (A < 0) { sign = sign * -1; }
```

```
    if (B < 0) { sign = sign * -1; }
```

```
    a = Math.abs(a);
```

```
    b = math.abs(b);
```

```
    int q = 0;
```

```
    int temp = 0;
```

```
    for (int i = 31; i >= 0; i--) {
```

```
        if (temp + (b << i) <= a) {
```

```
            temp = temp + (b << i);
```

```
            q = q + ((long)1 << i);
```

```
    }
```

```
    }
```

```
    if (sign < 0) { q = -q; }
```

```
    if (q > Integer.MAX_VALUE) {
```

```
        return Integer.MAX_VALUE;
```

```
    if (q < Integer.MIN_VALUE) {
```

```
        return Integer.MIN_VALUE;
```

T.C: O(32)

S.C: O(1)

3

return (int)j;



Tracing

long a = A;
long b = B;

```
int sign = 1;
if (a < 0) { sign = sign * -1; }
if (b < 0) { sign = sign * -1; }
a = Math.abs(a);
b = Math.abs(b);
```

int q = 0;
int temp = 0;

a: 95 b: 5 $\Rightarrow \text{ans} = 19$
 9 5
 1 4
 3 2 1 0
 visualisation: - - - 0 9 1 0 0 0 1 1
 $q = 0 + 16 + 2 = 18 + 1 = 19$

$$\text{temp} = 0 + 80 + 10 + 5$$

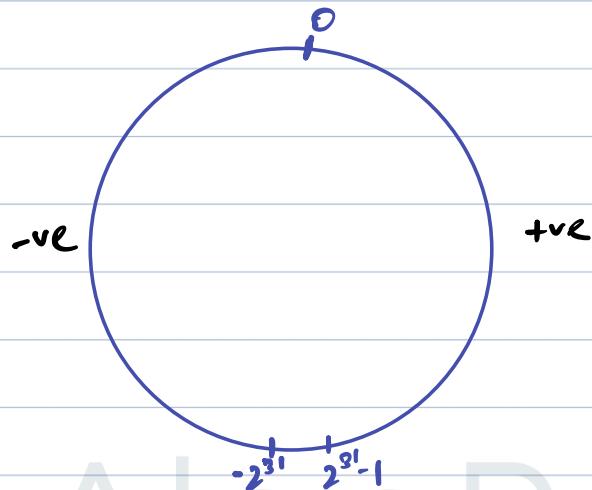
```
for (int i = 31; i >= 0; i--) {
  if (temp + (b << i) <= a) {
    temp = temp + (b << i);
    q = q + ((long)1 << i);
  }
}
```

3



* Numbers in DSA

int $\rightarrow \{-2^{31}, 2^{31}-1\}$



AlgoPrep



↳



AlgoPrep