for(i=1; i<n; i++) → runs n times → O(n)

for(i=1; i<n; i=i*2) → 1, 2, 4, 8, ... 2^k (k^th)

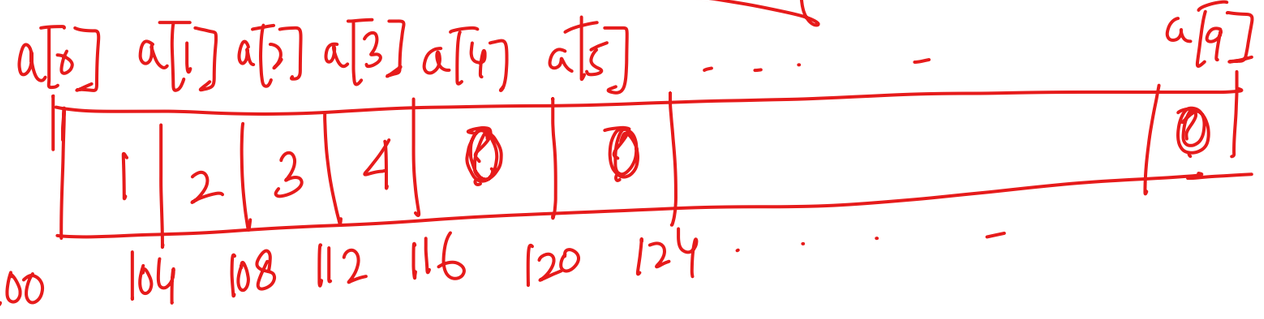
$2^k < n$

$\Rightarrow k < \log_2 n \Rightarrow$ runs $\log_2 n \rightarrow O(\log_2 n)$

int a[10];

An array is an linear Data structure which stores the homogeneous data elements in contiguous memory locations.

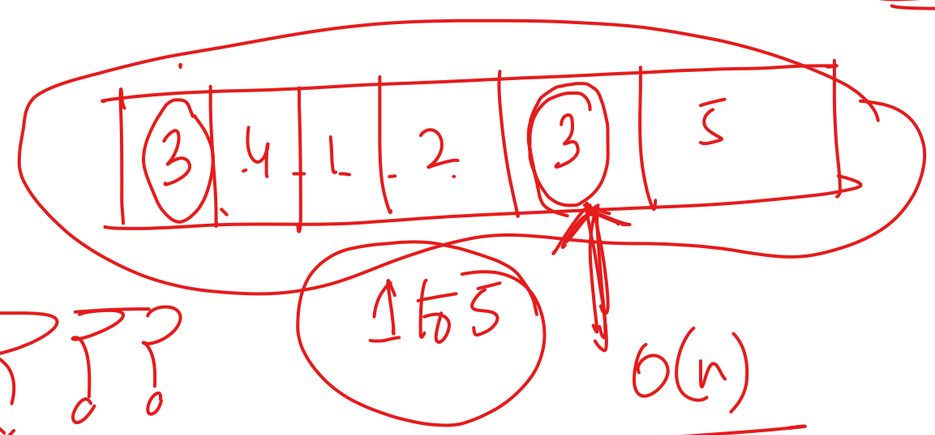
```
int a[10] = {1, 2, 3, 4};
```




```
int a[100] = {0};
```

Array is given. 1 to 100 each and every elements are present but unordered. one element has it duplicate value you have to find this duplicate value in O(n)

{ for(i=) } X



???

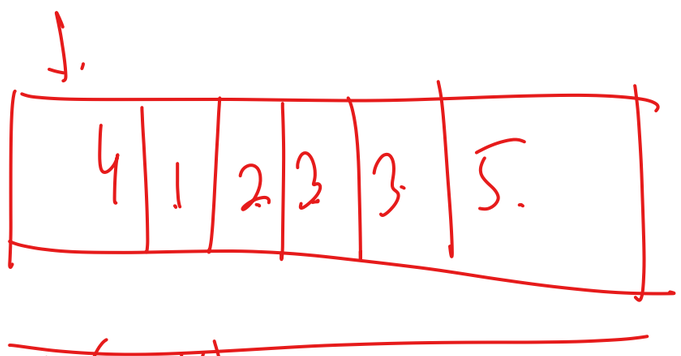
for(i=0; i<n; i++)

if(a[0]

}

if(a[0] == a[i])

if(a[i] == a[i+1])



$sum = \frac{n(n+1)}{2}$

$= \frac{5(5+1)}{2}$

$= 15$

O(nlogn)

x = a[i];

for(i=1; i<n; i++)

if(a[i]

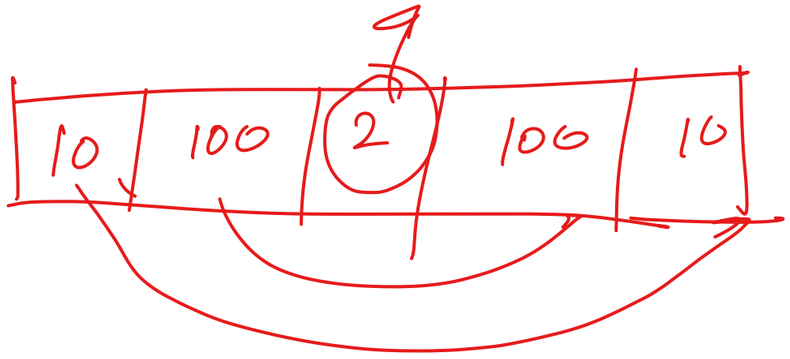
O(n)

for(i=0; i<n; i++)

sum = sum + a[i])

sum = 18

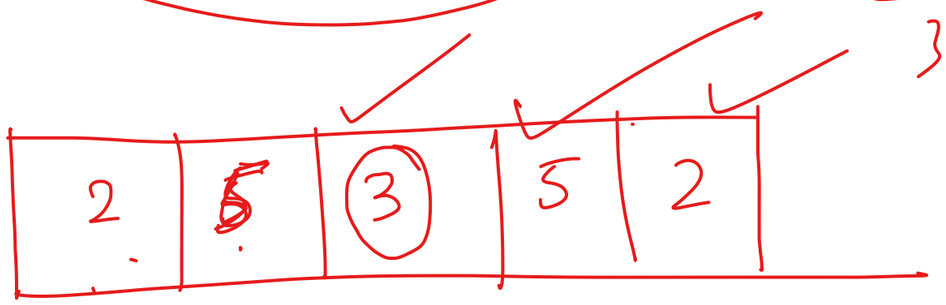
$dup = sum - Sum = 18 - 15 = 3$



each and every element has its duplicate value except one

$O(n)$

```
for(i=0; i<n; i++)
{  m = a[0];
}
for(
```



bitwise XOR operator. A=1, B=1 } ⇒0
A=0, B=0

$Y = \bar{A}B + A\bar{B}$

$= 1.0 + 0.1 = 0$

0 1 0
1 0 1
1 1 1
0 1 1
1 0 0
1 0 1
0 0 1
0 1 0
6 1 1 3

Complete

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}^n \text{ in } \boxed{O(1)} \text{ time.}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}^n = \begin{bmatrix} 1 & n & n(n+1)/2 \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix}$$

```
for (i=0; i<n; i++)
  { for (j=0; j<n; j++)
      scanf ("%d", &a[i][j]);
  }
```

$O(n^2)$

1, 3, 6

$\frac{n(n+1)}{2}$

Array with pointers

(Random Access)

```
int a[5] = {10, 20, 30, 40, 50};
```

a[0] a[1] a[2] a[3] a[4]

10	20	30	40	50

100 104 108 112 116

let, int take 4 byte

✓ $a[3] = *(a + 3 * \text{data byte})$

$= *(100 + 3 * 4)$

$= *$ to 112

= value at 112

= 40

$a[i] = *(a + i * D.size)$

$= *(i * size + a)$

$= i[a]$

```
int main ()
{ char s[ ] = " DEBAYAN ";
  int i;
  for (i=0; s[i] != '\0' ; i++)
    printf ("\n %c %c %c %c", s[i], *(s+i), *(i+s), i[s]);
  return 0;
}
```

D	E	B	A	Y	A	N	\0

o/p

D D D D
E E E E
.
.
.
N N N N

5 [" INTERVIEW "]
  0 1 2 3 4 5

a[5]

o/p :- 'V'

$$a[i] = *(a+i)$$

$$= *(i+a)$$

$$= i[a]$$

$$2+3 \equiv 3+2$$

$$*(a+i) = a[i]$$

$$*(i+a) = i[a]$$