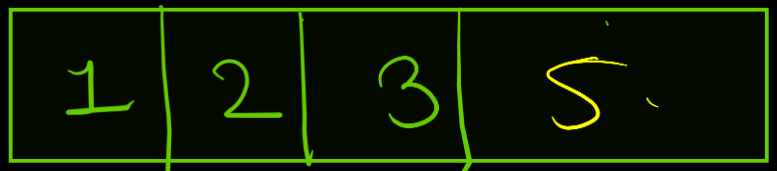
Q 2. print odd betn 2 to n. & if num > 10, check it's digits

n-1

1 to n-1

↳ 4 miss



→ 1 to n-1 (4)

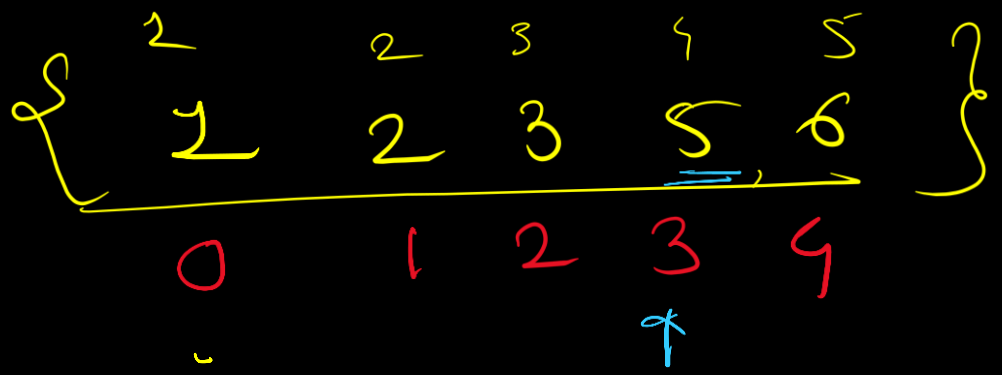
1 to n (5)

arr[i] != i+1

arr[3] != 3+1 ⇒ 5 != 4

}

Arrays.sort( )



$\Rightarrow 1 \longrightarrow n$.

1 2 3 4 5 6

0 1 2 3 4 5

{1, 2, 3, 4} = n = 1, 2, 3, 4, 5

0 1 2 3

i = 1

1 2 3 4 → 5

O(nlogn) ⇒

further optimization.

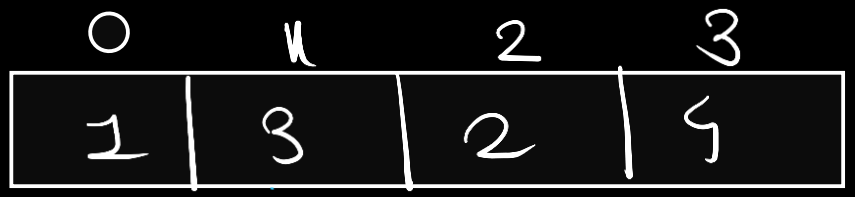
medium

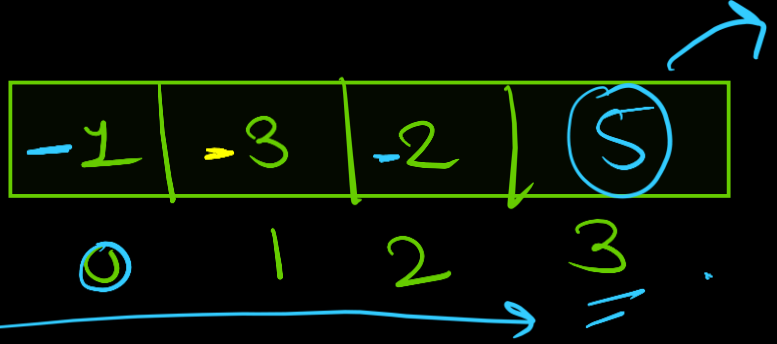
pattern.

↳ 1 to n → Sort →

↳ indexes →

{ 1, 3, 2, { }



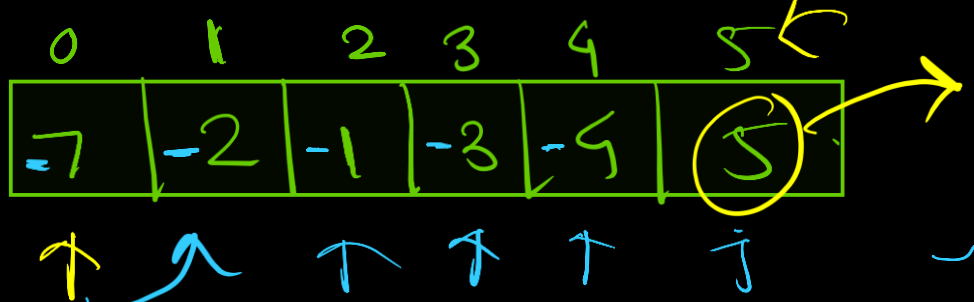


⇒ +, −

divide.

$|-2| = 2 - 1 = 1$.

$-arr[arr[i]-1] =$



$3 - 1 = 2$

idx >= size.

Cont.

int idx = arr[i] - 1

arr[i] = -2 * arr[i]

3) don't modify array

⇒ Maths.

1+3+4+5+6

19

1	3	4	5	6

= 19 − 5

1 to n Sum. 1st n natural number sum.

$$6 = \frac{6 * (7)}{2} = 42/2 = 21 \text{ Sum}$$

21 − 19 = 2

1 to n = $\frac{n*(n+1)}{2}$ − Σ arr = missing elem

tc: O(n)

Q.1 array {4, 6, 7, 9} ⇐ missing

9	7	6	5

= 8

5	6	7	9

⇒ arr[i] == arr[i+1]

5 6-1

7 != 8 9-1 arr[i+1] -1

```java
// User function Template for Java
class Solution {
    int missingNumber(int arr[]) {
        // code here
        Arrays.sort(arr);
        for(int i = 0; i < arr.length; i++){
            if(arr[i] != i+1)return i+1;
        }
        return arr.length + 1;
    }
}
```

```java
// User function Template for Java
class Solution {
    int missingNumber(int arr[]) {
        int n = arr.length + 1;
        int sum = 0;
        for(int i = 0; i < arr.length; i++){
           sum += arr[i];
        }
        int actualSum = (n * (n + 1))/2;
        return actualSum - sum;
    }
}
```

```java
// User function Template for Java
class Solution {
    int missingNumber(int arr[]) {
        for(int i = 0; i < arr.length; i++){
            int idx = Math.abs(arr[i]) - 1;
            if(idx >= arr.length)continue;
            arr[idx] *= -1;
        }
        for(int i= 0; i < arr.length; i++)
        {
            if(arr[i] > 0)return i+1;
        }
        return arr.length + 1;
    }
}
```

Q 2

n

↳ odd return.

if ele > 10

↳ digits sum = odd.

n = 15

1 3 5 7 9 10 12 14 16

2345 = digit Sum

2+3+4+5 = 14

$2345/10$ =

234 ⇒ 23

```
      234
   10)2345
      20
      ---
       34
       30
       --
        45
        40
        --
         5
```

num%10 = lastDigit

num/10 = Quotient

↳ new number

5 → (5)

Sum

```
while(num > 0){
    dig = num % 10
    num = num / 10
    sum += dig
}
```

num = 0

sum = 5 + 4 + 3

2 = 14

```java
public class PrintOddTCS {
    new *
    public static void main(String[] args) {
        int n = 15;

        for(int i = 1; i <= n; i++){
            if(i > 9){
                int num = i;
                int sum = 0;
                while(num > 0){
                    sum += (num % 10);
                    num = num / 10;
                }
                if((sum % 2 ) == 1) System.out.print(i + " ");
            }
            else{
                if((i % 2 ) == 1)System.out.print(i + " ");
            }
        }
    }
}
```