Rotate array

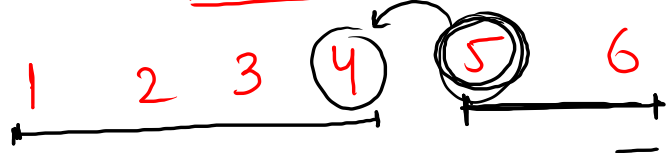
arr 1 2 3 4 5 6 , k = 2

k=1, 6 1 2 3 4 5

k=2, 5 6 1 2 3 4

gmp

k = 2, n = 6

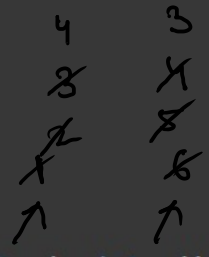


step1 1 2 3 4 6 5 (n-k, n-1)

step2 4 3 2 1 6 5 (0, n-k-1)

step3 5 6 1 2 3 4 (0, n-1)

```java
public void rotate(int[] arr, int k) {
    int n = arr.length;
    k = k % n;

    // step 1
    reverse(arr, n - k, n - 1);

    // step 2
    reverse(arr, 0, n - k - 1);

    // step 3
    reverse(arr, 0, n - 1);
}

public void reverse(int[] arr, int si, int ei) {
    while ( si < ei ) {
        swap(arr, si, ei);
        si++;
        ei--;
    }
}

public void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

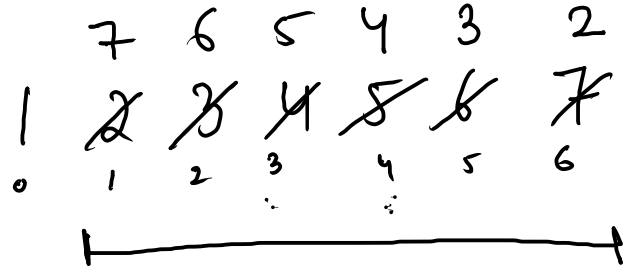


[1,2,3,4,5,6,7], k = ~~3~~ 17%7 = 3

Step1) 1 2 3 4 7 6 5 (reverse k elements from last)

Step2) 4 3 2 1 7 6 5 (reverse the remaining)

Step3) 5 6 7 1 2 3 4 (reverse the whole array)



K = 3

K = K % n

= 3 % 7 = 3

```java
public void rotate(int[] nums, int k) {
    int n = nums.length;
    k = k % n;

    // step 1
    reverse(arr, n - k, n - 1);

    // step 2
    reverse(arr, 0, n - k - 1);

    // step 3
    reverse(arr, 0, n - 1);
}
```

arr = 1 2 3 4 5

k=1, 5 1 2 3 4

k=2, 4 5 1 2 3

k=3, 3 4 5 1 2

k=4, 2 3 4 5 1

k=5, 1 2 3 4 5

n=5

k=6 %.5 = 1

k=7 %.5 = 2

k=8 %.5 = 3

k=9 %.5 = 4

k=10 %.5 = 0

k=11 %.5 = 1

k=12 %.5 = 2

k=13 %.5 = 3

k=14 %.5 = 4

k=15 %.5 = 0

Quel Sort 01

arr	0	1	0	1	1	0	0	1	0	0	1
↳	0	0	0	0	0	0	1	1	1	1	1

T.C = O(n)

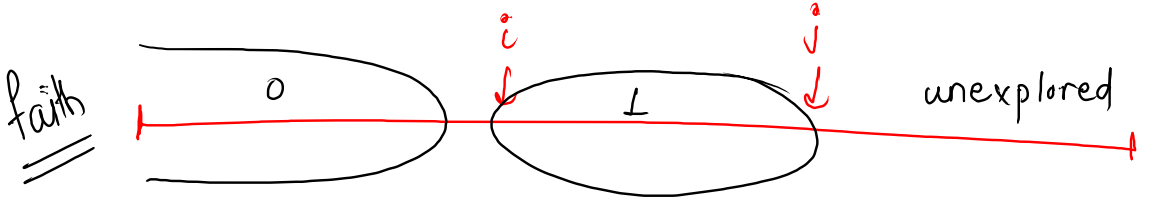
S.C = O(1)

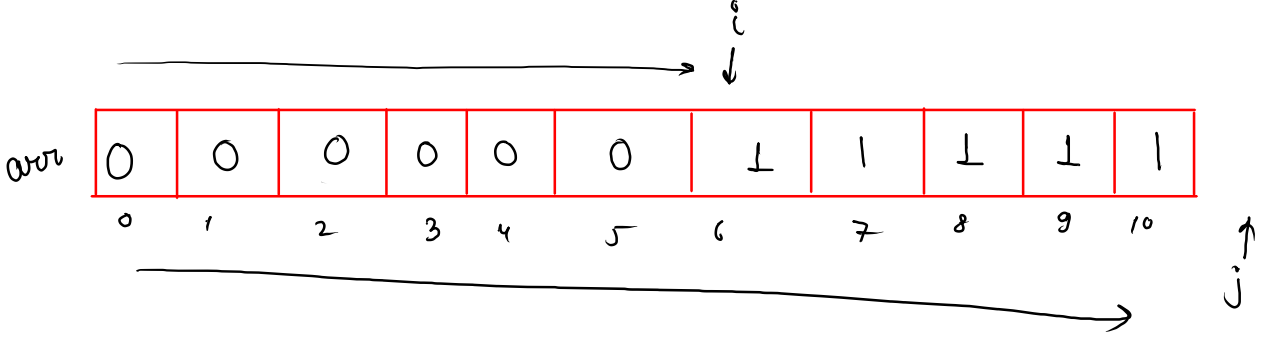
arr	0	1	0	1	1	0	0	1	0	0	1

int i, j ;

meaning of i ⇒ we will have all 0's before i

meaning of j ⇒ we will have all 1's before j and after i





check j each time

if j == 1

j++

else swap(i, j)

i++

j++

T.C = O(N)

S.C = O(1)

```java
public static void sort01(int[] arr, int n) {
    int i = 0; // all elemets before i will be 0
    int j = 0; // all elements before j will be 1
    while (j < arr.length) {
        if ( arr[j] == 1 ) {
            j++;
        } else {
            swap(arr, i, j);
            i++;
            j++;
        }
    }

    for (int k = 0; k < arr.length; k++) {
        System.out.print(arr[k] + " ");
    }
}

public static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

Sort 012

0	1	1	0	2	1	0	2	0	1	1	0

T.C = O(N)

S.C = O(1)

i , j , k