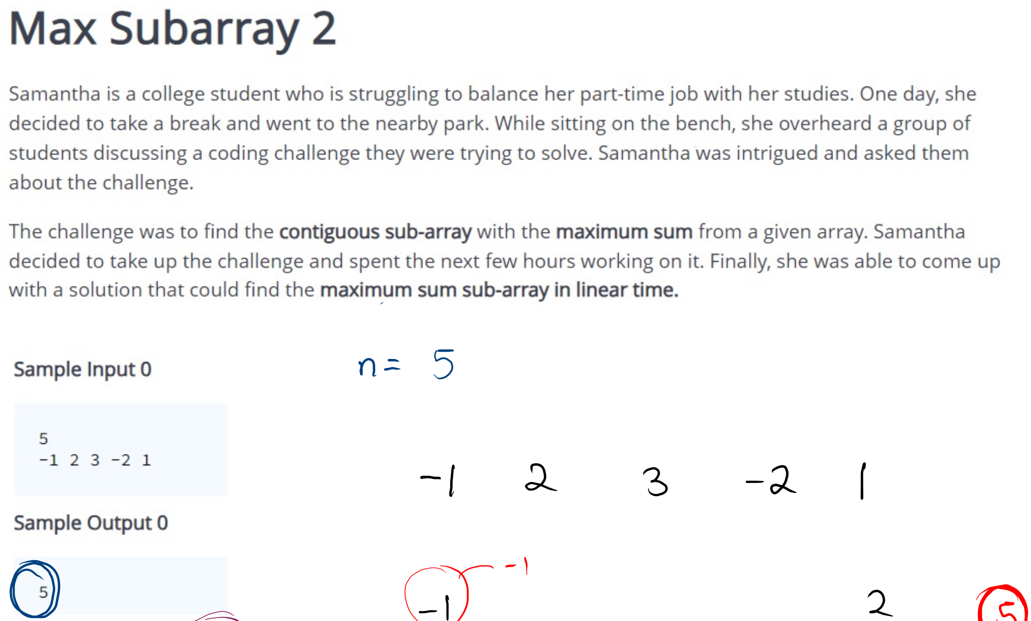


Max Subarray 2

Samantha is a college student who is struggling to balance her part-time job with her studies. One day, she decided to take a break and went to the nearby park. While sitting on the bench, she overheard a group of students discussing a coding challenge they were trying to solve. Samantha was intrigued and asked them about the challenge.

The challenge was to find the **contiguous sub-array** with the **maximum sum** from a given array. Samantha decided to take up the challenge and spent the next few hours working on it. Finally, she was able to come up with a solution that could find the **maximum sum sub-array in linear time.**

Sample Input 0

```
5
-1 2 3 -2 1
```

Sample Output 0

```
5
```

O(n)

n = 5

-1 2 3 -2 1

-1 → -1

-1 2 → 1

-1 2 3 → 4

-1 2 3 -2 → 2

-1 2 3 -2 1

2

2 3 → 5

2 3 -2

2 3 -2 1

3

3 -2

3 -2 1

-2

-2 1

1

TLE

1. print all subarray
2. find sum of all sub-array → individually.
3. solve → max(sum).

} $O(n^3)$ →

1 → loop → st

2 → loop → end } (n)

3 → st — end.

⇒ $O(n^3)$

Business man → option 1 → old in profit

acquire

→ option 2

start a new business

(profit) sum = 0

(ans). max = ~~-∞~~ ~~1~~ ~~2~~ 5

-1	2	3	-2	1
0	1	2	3	4

sum = ~~1~~ ~~2~~ ~~5~~ ~~3~~ 4

i

**

psum > 0 +ve

add. A[i]

3 + 1 = 4

$\boxed{1}$

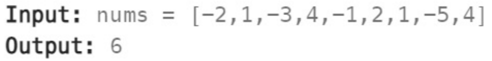
prev sum = ~~0~~ ~~-1~~ ~~2~~ ~~5~~ ~~3~~ 4

max = ~~-∞~~ ~~-1~~ ~~2~~ 5

-1 2 3 -2 1

i

2 > 0



```
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
```

*** Kadane's Algo

psum = ~~0~~ ~~-2~~ ~~1~~ ~~-2~~ ~~4~~ ~~3~~ ~~5~~ ~~6~~ ~~1~~ 5

max = ~~-∞~~ ~~-2~~ ~~1~~ ~~4~~ ~~5~~ 6

-2 1 -3 4 -1 2 1 -5 4

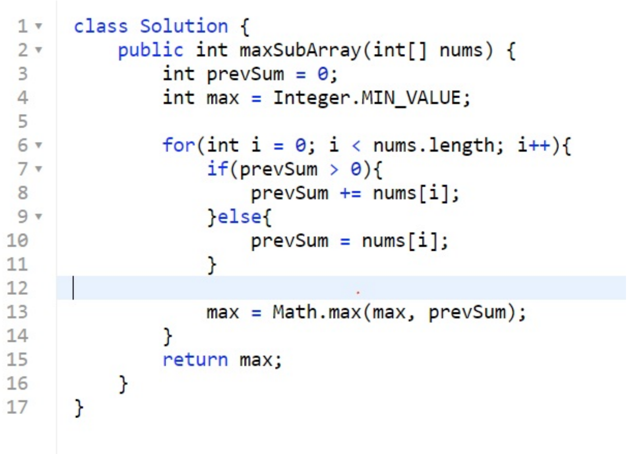
i

psum > 0

↳ psum += A[i]

else

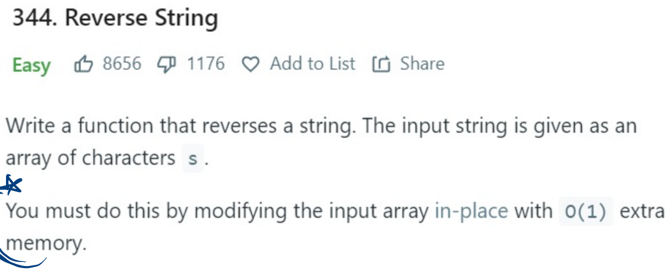
psum = f ·]



```
class Solution {
    public int maxSubArray(int[] nums) {
        int prevSum = 0;
        int max = Integer.MIN_VALUE;

        for(int i = 0; i < nums.length; i++){
            if(prevSum > 0){
                prevSum += nums[i];
            }else{
                prevSum = nums[i];
            }

            max = Math.max(max, prevSum);
        }
        return max;
    }
}
```



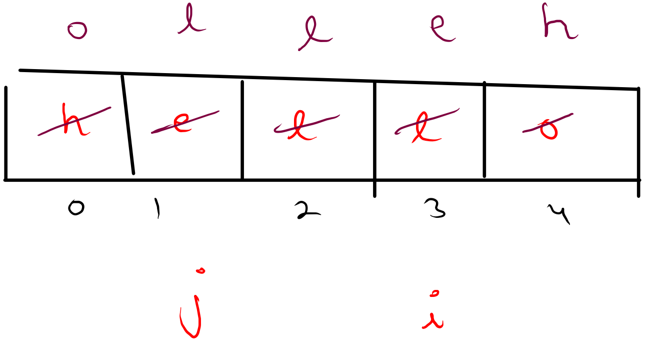
344. Reverse String

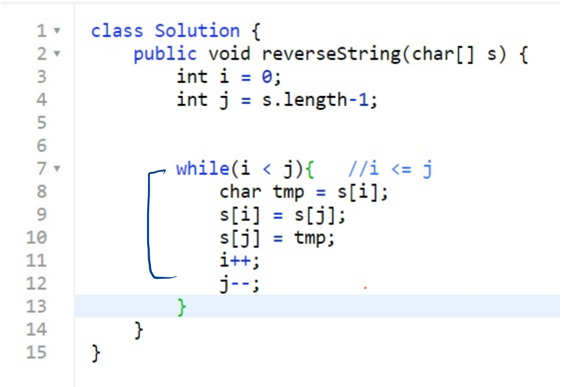
Easy 8656 1176 Add to List Share

Write a function that reverses a string. The input string is given as an array of characters `s`.

*You must do this by modifying the input array in-place with `O(1)` extra memory.

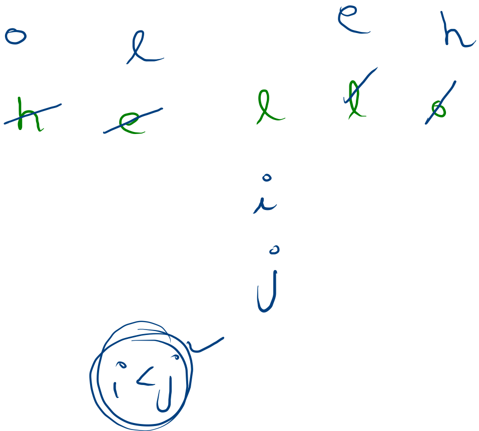
? ?

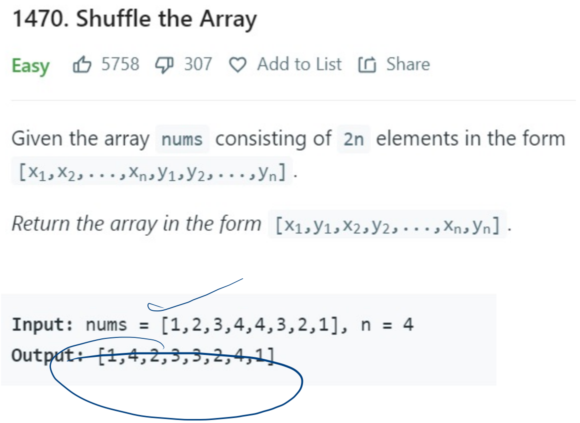




```
class Solution {
    public void reverseString(char[] s) {
        int i = 0;
        int j = s.length-1;


        while(i < j){   //i <= j
            char tmp = s[i];
            s[i] = s[j];
            s[j] = tmp;
            i++;
            j--;
        }
    }
}
```





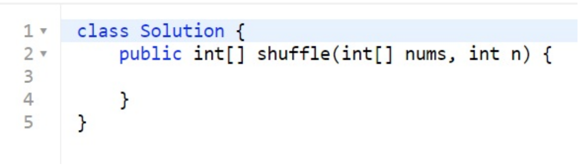
1470. Shuffle the Array

Easy 5758 307 Add to List Share

Given the array `nums` consisting of `2n` elements in the form $[x_1,x_2,...,x_n,y_1,y_2,...,y_n]$.

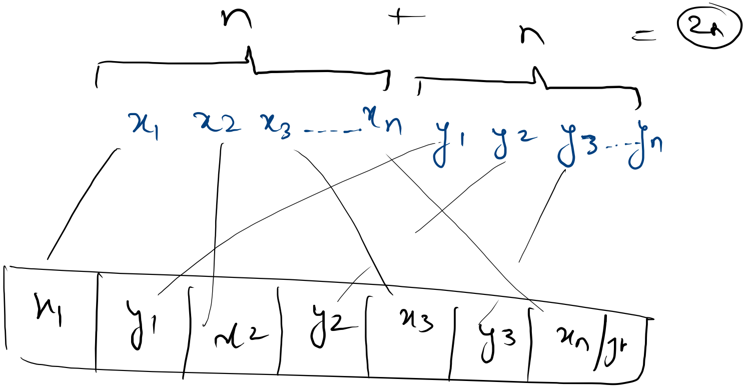
Return the array in the form $[x_1,y_1,x_2,y_2,...,x_n,y_n]$.

```
Input: nums = [1,2,3,4,4,3,2,1], n = 4
Output: [1,4,2,3,3,2,4,1]
```



```
class Solution {
    public int[] shuffle(int[] nums, int n) {

    }
}
```



1 2 3 4 4 3 2 1 → 1 1 2 2 3 3 4 4

$n = 4$

nums →	1	2	3	4	4	3	2	1
	0	1	2	3	4	5	6	7

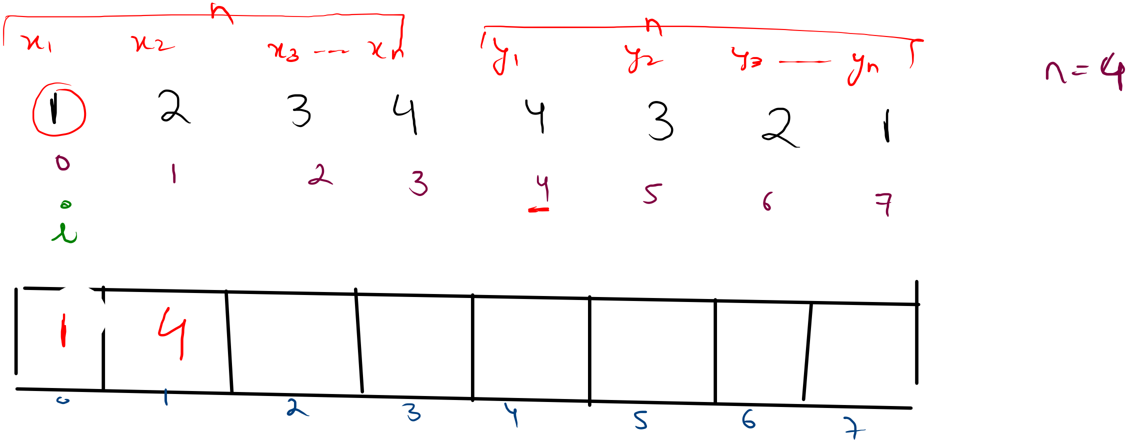
i

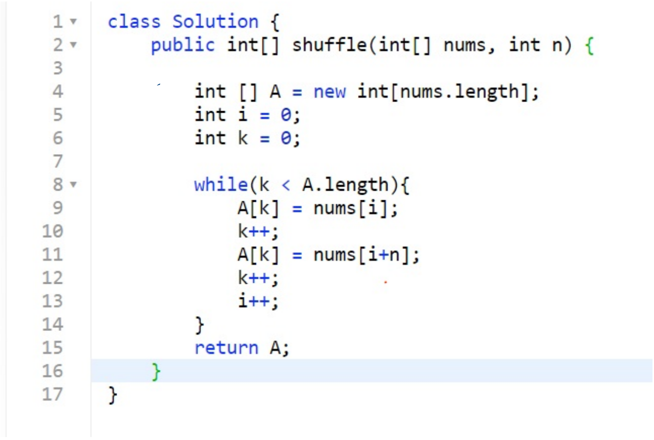
new A	1	4	2	3	3	2	4	1
	0	1	2	3	4	5	6	7

K

A[k] = nums[i]

A[k] = nums[i+n]

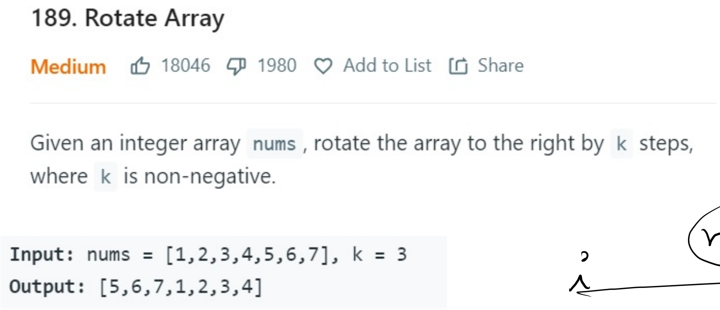




```
class Solution {
    public int[] shuffle(int[] nums, int n) {

        int [] A = new int[nums.length];
        int i = 0;
        int k = 0;

        while(k < A.length){
            A[k] = nums[i];
            k++;
            A[k] = nums[i+n];
            k++;
            i++;
        }
        return A;
    }
}
```

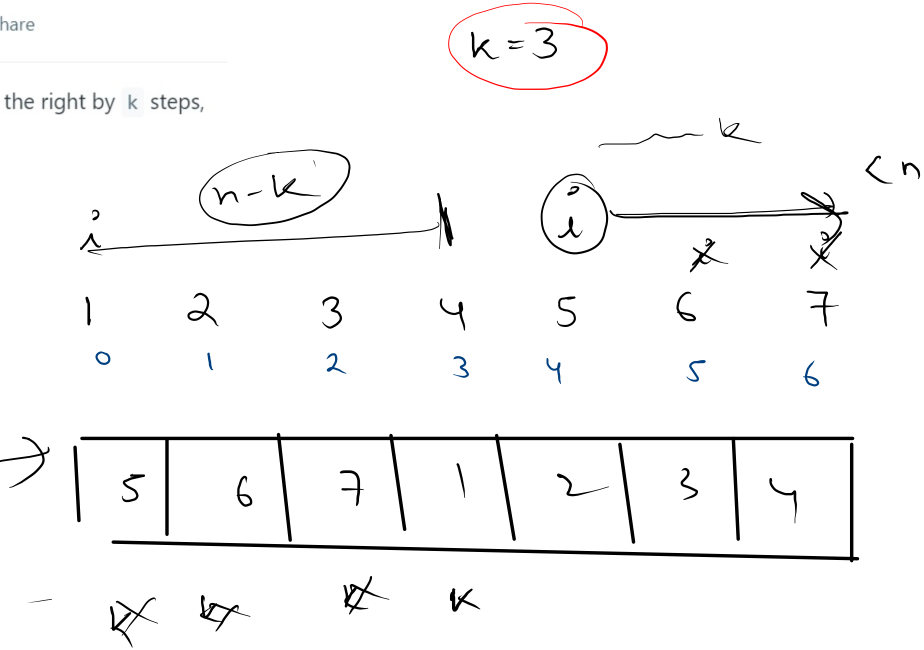


189. Rotate Array

Medium 18046 1980 Add to List Share

Given an integer array `nums`, rotate the array to the right by `k` steps, where `k` is non-negative.

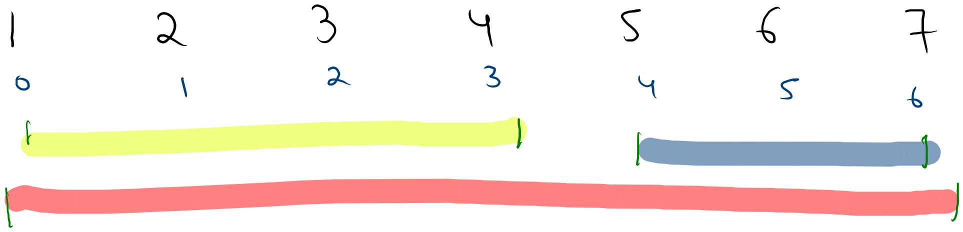
```
Input: nums = [1,2,3,4,5,6,7], k = 3
Output: [5,6,7,1,2,3,4]
```



SC => O(n)

TC => O(n)

k=3



4 3 2 1 7 6 5

5 6 7 1 2 3 4 ans.

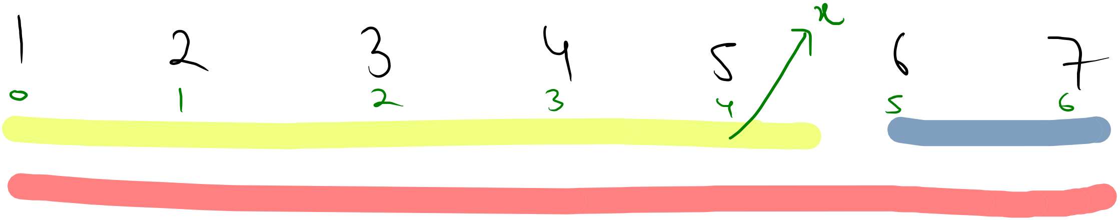
reverse y → O(n)

reverse b → O(n)

reverse r → O(n)

$3n = \boxed{O(n)}$

k = 2



complete

start end

reverse (A 0 x)

reverse (A x+1 n-1)

reverse (A 0 n-1)

6 7 1 2 3 4 5

find x ?

eg. $k=3$ 1 2 3 4 5 6 7 ⇒ 5 6 7 1 2 3 4

$k=4$ 1 2 3 4 5 6 7 ⇒ 4 5 6 7 1 2 3

$k=5$ 1 2 3 4 5 6 7 ⇒ 3 4 5 6 7 1 2

$k=6$ 1 2 3 4 5 6 7 ⇒ 2 3 4 5 6 7 1

complete

reverse (A $\underline{0}$ $\underline{x}$)

reverse (A $\underline{x+1}$ $\underline{n-1}$)

reverse (A $\underline{0}$ $\underline{n-1}$)