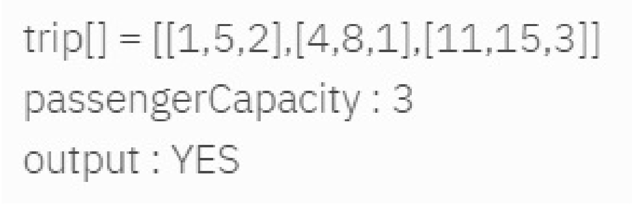
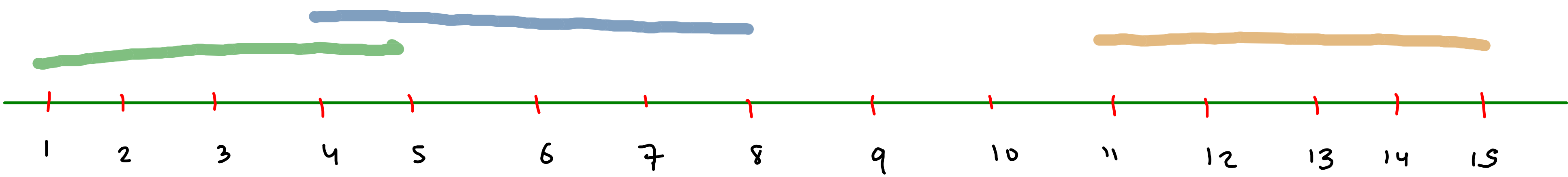
1094. Car Pooling

trip -> [from, to, no. of passenger]



```
trip[] = [[1,5,2],[4,8,1],[11,15,3]]
passengerCapacity : 3
output : YES
```

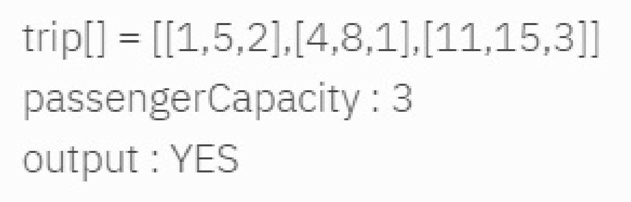


1 - 4 -> 2

4 - 5 -> 3

5 - 8 -> 1

11 - 15 -> 3



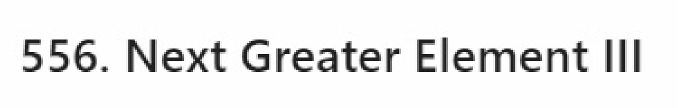
trip[] = [[1,5,2],[4,8,1],[11,15,3]]

passengerCapacity : 3

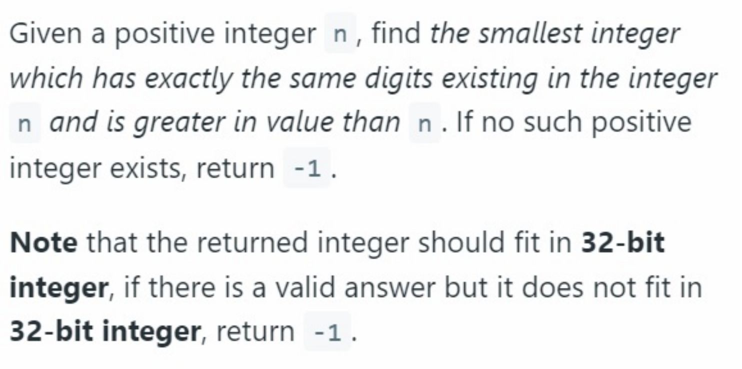
output : YES

Range addition

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
arr	2	0	0	1	-2	0	0	-1	0	0	3	0	0	0	-3
PS	2	2	2	3	1	1	1	0	0	0	3	3	3	3	0



556. Next Greater Element III



Given a positive integer `n`, find *the smallest integer which has exactly the same digits existing in the integer* `n` *and is greater in value than* `n`. If no such positive integer exists, return `-1`.

Note that the returned integer should fit in **32-bit integer**, if there is a valid answer but it does not fit in **32-bit integer**, return `-1`.

n = 1 3 2

1 3 2

1 2 3

2 1 3

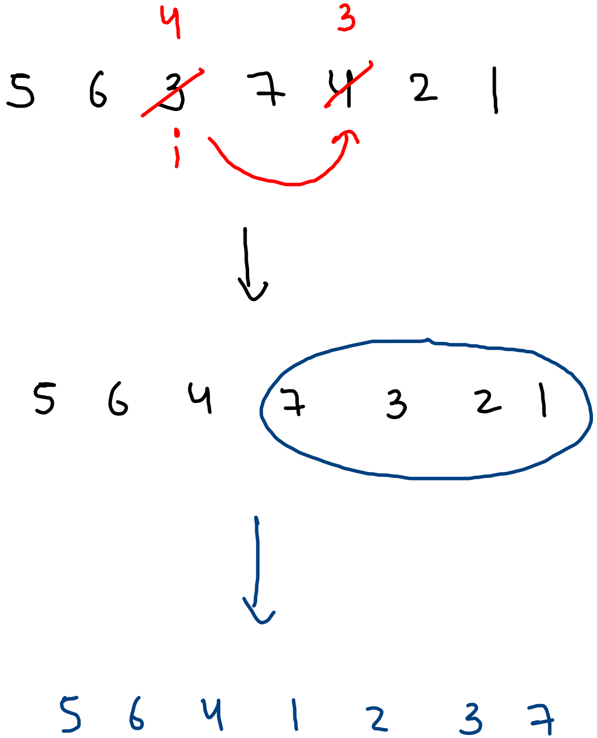
2 3 1

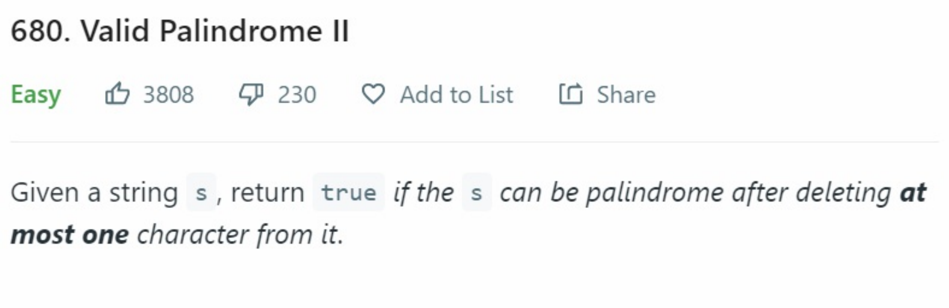
3 1 2

3 2 1

n = 4 3 2 1

ans = -1


4
3
5 6 ~~3~~ 7 ~~4~~ 2 1
i
5 6 4 7 3 2 1
5 6 4 1 2 3 7



680. Valid Palindrome II

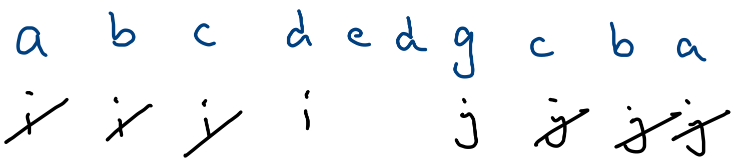
Easy 3808 230 Add to List Share

Given a string `s`, return `true` *if the* `s` *can be palindrome after deleting **at most one** character from it.*

opr : 0, 1

if delete ith char

→ (i+1) to j pal



if delete jth char

→ i to j-1 pal

Wiggle Sort 1

2. Reorder it in-place such that : arr[0] <= arr[1] >= arr[2] <= arr[3]

indexes 0 $\le$ 1 $\ge$ 2 $\le$ 3 $\ge$ 4 $\le$ 5 . . . .

Input: [3, 5, 2, 1, 6, 4]

Output: [1, 6, 2, 5, 3, 4]

c b

a $\le$ ~~b~~ $\ge$ ~~c~~ d e f

0 1 2 3 4 5

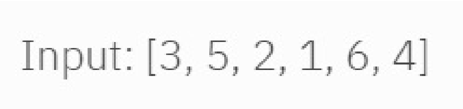
i

a < b

b < c

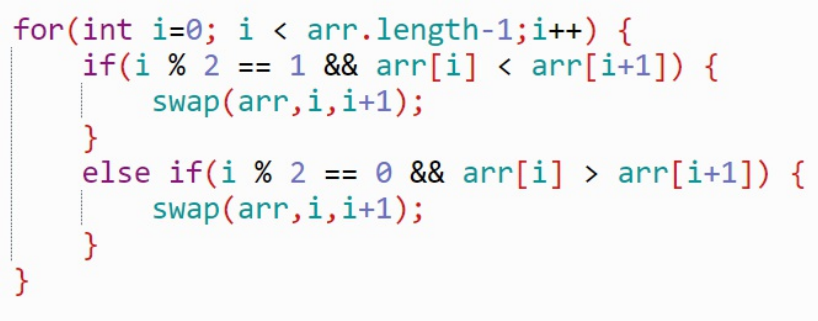
odd index	even index
is greater or equal with left & right	is smaller or equals to left & right.

`if (arr[i] < arr[i+1]){`
`swap (i, i+1);`
`}`
	`if (arr [i] > arr [i+1]) {`
	`swap (i, i+1);`
	`}`

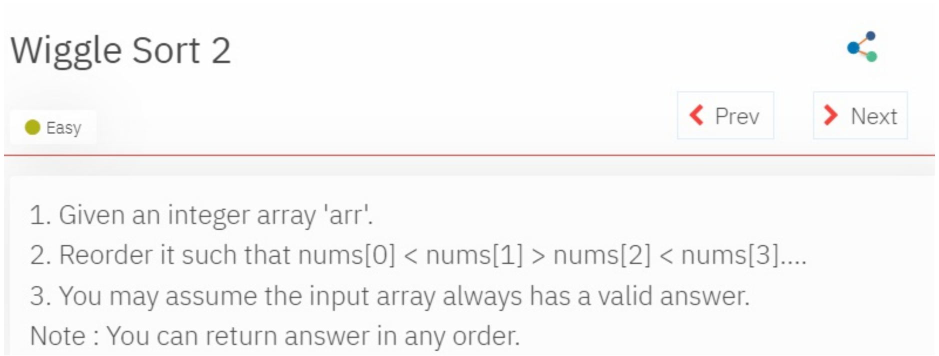


Input: [3, 5, 2, 1, 6, 4]

$3_0 \le 5_1 \ge 1_2 \le 6_3 \ge 2_4 \le 4_5$



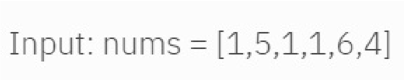
```
for(int i=0; i < arr.length-1;i++) {
    if(i % 2 == 1 && arr[i] < arr[i+1]) {
        swap(arr,i,i+1);
    }
    else if(i % 2 == 0 && arr[i] > arr[i+1]) {
        swap(arr,i,i+1);
    }
}
```



Wiggle Sort 2

1. Given an integer array 'arr'.
2. Reorder it such that nums[0] < nums[1] > nums[2] < nums[3]....
3. You may assume the input array always has a valid answer.
Note : You can return answer in any order.

index 0 < 1 > 2 < 3 > 4 < 5 . . . . . .



Input: nums = [1,5,1,1,6,4]

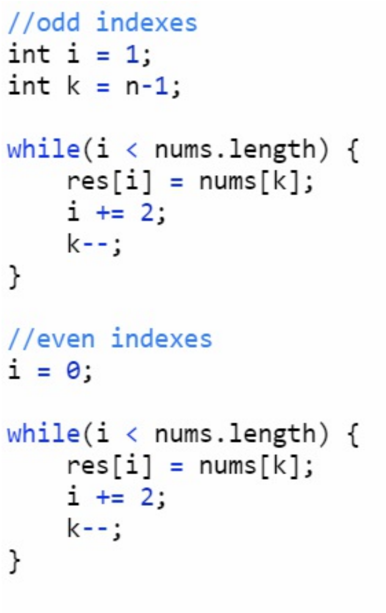
after sorting

a	b	c	d	e	f
0	1	2	3	4	5

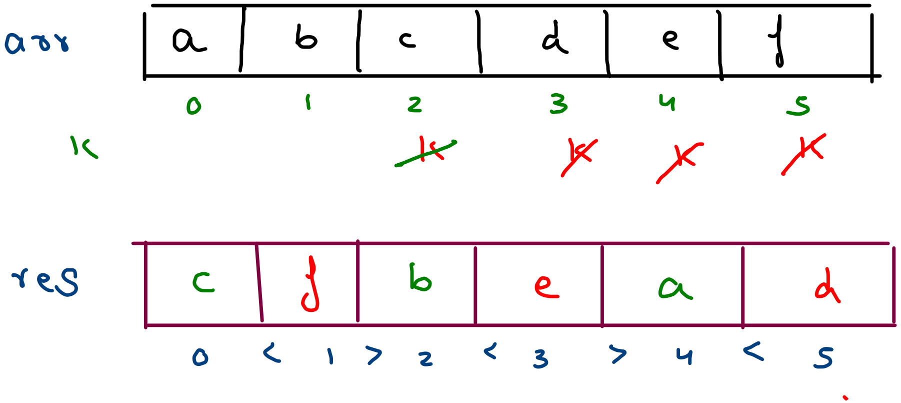
a < b < c < d < e < f

ans:

c	f	b	e	a	d
0 <	1 >	2 <	3 >	4 <	5



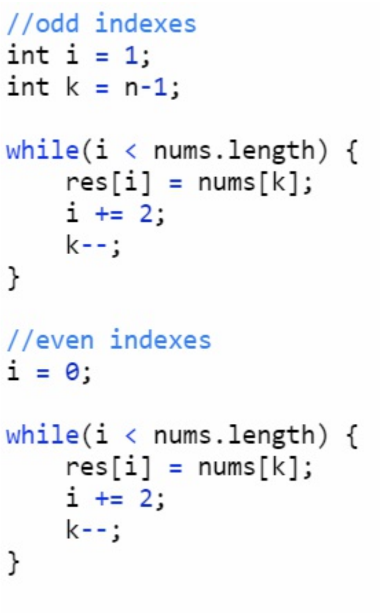
```
//odd indexes
int i = 1;
int k = n-1;

while(i < nums.length) {
    res[i] = nums[k];
    i += 2;
    k--;
}

//even indexes
i = 0;

while(i < nums.length) {
    res[i] = nums[k];
    i += 2;
    k--;
}
```



arr	a	b	c	d	e	f
	0	1	2	3	4	5
k			~~k~~	~~k~~	~~k~~	~~k~~

res	c	f	b	e	a	d

0 < 1 > 2 < 3 > 4 < 5



```
//odd indexes
int i = 1;
int k = n-1;

while(i < nums.length) {
    res[i] = nums[k];
    i += 2;
    k--;
}

//even indexes
i = 0;

while(i < nums.length) {
    res[i] = nums[k];
    i += 2;
    k--;
}
```

arr	a	b	c	d	e	f	g
	0	1	2	3	4	5	6

res	d	g	c	f	b	e	a
	0	1	2	3	4	5	6