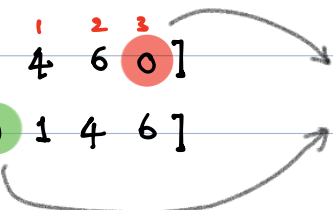
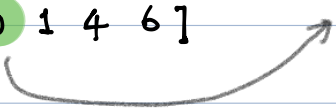


Question: Given an array of integers, where elements from $(0 \text{ to } N-1)^{\text{th}}$ index is sorted except the last element sort the array such the last index element moves to its correct position.

array = $[1^0, 4^1, 6^2, 0^3]$  last index element is not sorted;
output = $[0, 1, 4, 6]$  last index element moved to its correct position.

Solution

Run a loop and compare i^{th} element with $i^{\text{th}}+1$ element, if i^{th} element is greater than $i^{\text{th}}+1$, swap them

Note: as the unsorted element is at last index, start from last index.

Iteration -1

array = $[1^0, 4^1, 6^2, 0^3]$

$\text{arr}[2] > \text{arr}[3]$, yes; swap them

$\text{temp} = \text{arr}[3]$

$\text{arr}[3] = \text{arr}[2]$

$\text{arr}[2] = \text{temp}$

array = $[1^0, 4^1, 0^2, 6^3]$

Iteration -2

$arr[1] > arr[2]$, Yes; swap them

$temp = arr[2]$

$arr[2] = arr[1]$

$arr[1] = temp$

array = [1⁰ 0¹ 4² 6³]

Iteration -3

$arr[0] > arr[1]$ Yes; swap them

$temp = arr[1]$

$arr[1] = arr[0]$

$arr[0] = temp$

array = [0⁰ 1¹ 4² 6³]

code

```
def sort(arr):
```

```
    n = len(arr)
```

```
    for i in range(n-2, -1, -1):
```

```
        if arr[j] > arr[j+1]:
```

```
            temp = arr[j]
```

```
            arr[j] = arr[j+1]
```

```
            arr[j+1] = temp
```

```
        else:
```

```
            break
```

$j = (4-2) = 2$ $j+1 = 3$

$arr[2] > arr[3]$

$j = 2-1 = 1$ $j+1 = 1+1 = 2$

$j = 1-1 = 0$ $j+1 = 0+1 = 1$