📆 Day 01 – Arrays: Core Concepts + Sorting

🔁 Array Rotation – Left

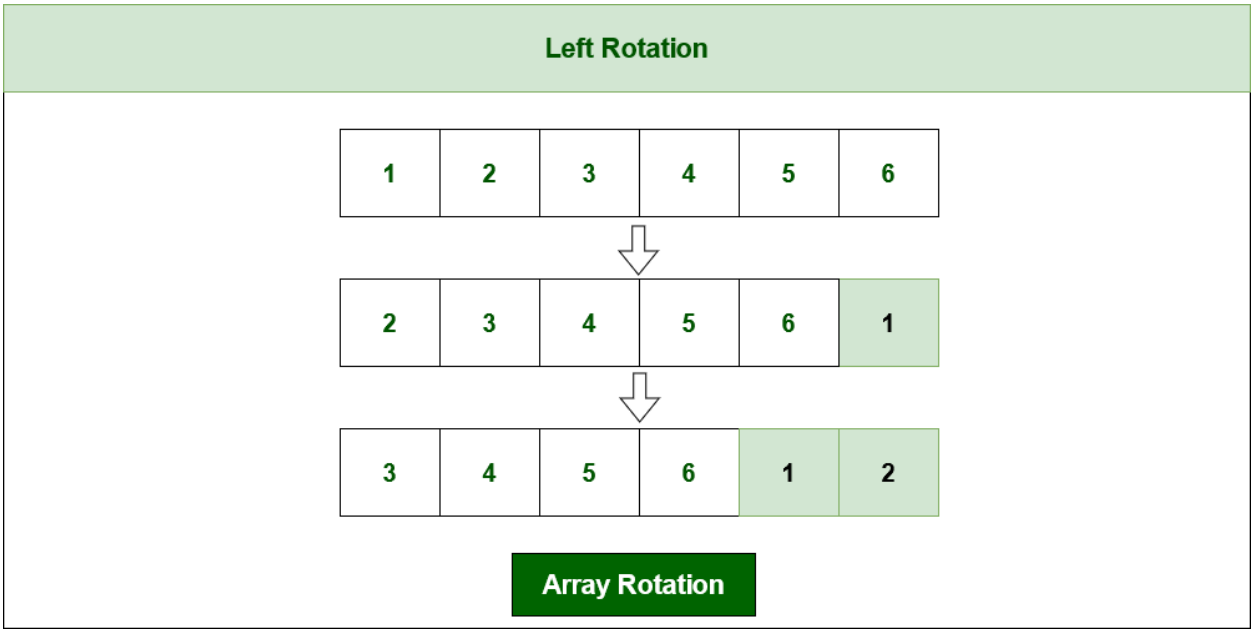
🧠 Logic:

- Remove the first element and append it to the end.
- Can be repeated `k` times for `k-left rotations`.

📦 Code (Python):

```
python
CopyEdit
def left_rotate(arr):
    temp = arr[0]
    for i in range(1, len(arr)):
        arr[i-1] = arr[i]
    arr[-1] = temp
```

🖼️ Diagram (Left Rotation):



🔄 Array Rotation – Right

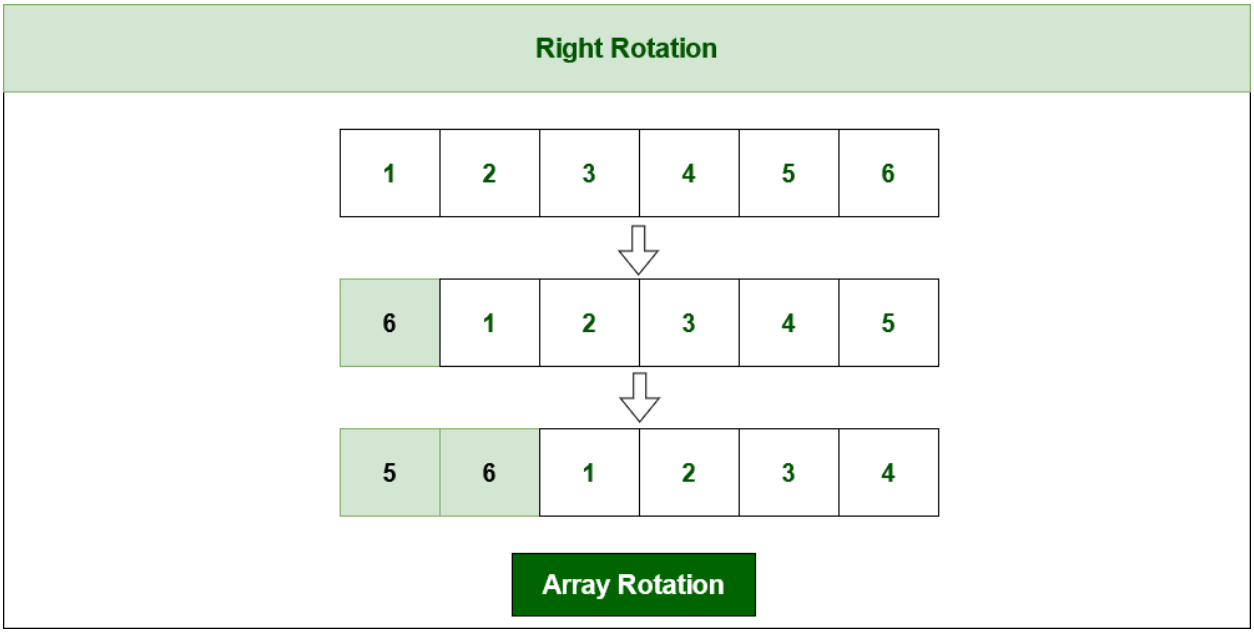
🧠 Logic:

- Take the last element and place it at the beginning.
- Can be repeated `k` times for `k-right rotations`.

📦 Code (Python):

```
python
CopyEdit
def right_rotate(arr):
    temp = arr[-1]
    for i in range(len(arr)-2, -1, -1):
        arr[i+1] = arr[i]
    arr[0] = temp
```

🖼️ Diagram (Right Rotation):



🔃 Bubble Sort

🧠 Logic:

- Compare adjacent pairs, swap if needed
- Continue for all passes
- Time Complexity: O(n^2)

📦 Code (Python):

```
python
CopyEdit
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

🖼️ **Diagram (Bubble Sort Flow):**

🎯 Selection Sort

🧠 **Logic:**

- Find the minimum element and move it to the front
- Repeat for remaining unsorted portion
- Time Complexity: $O(n^2)$

📦 **Code (Python):**

```
python
CopyEdit
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

🖼️ **Diagram (Selection Sort Flow):**

Selection Sort in ascending order