Rotate Array

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Reverse
language	C++
solved on	@18/11/2024
link	https://www.geeksforgeeks.org/problems/rotate-array-by-n-elements-1587115621/1
Completion	☑

Intuition

The problem requires rotating an array by `d` elements. Instead of moving elements one by one, we can use a more efficient approach by reversing parts of the array. The intuition behind this method is to divide the array into segments, reverse each segment, and finally reverse the entire array to achieve the desired rotation.

Approach

1. **Modulo operation**: Take `d % arr.size()` to handle cases where `d` is greater than the size of the array, ensuring no unnecessary rotations.
2. **Reverse segments**:
   - Reverse the first `d` elements to bring them in their reversed order.
   - Reverse the remaining elements from `d` to the end of the array.
3. **Reverse the entire array**: This step combines the two reversed segments to achieve the rotation effect.

This approach effectively rotates the array in place without needing extra space.

Complexity

Time Complexity:

- **Reversal operations**: Each reversal takes `O(k)` time, where `k` is the length of the subarray.
  - First reversal: `O(d)`
  - Second reversal: `O(n - d)`
  - Final reversal: `O(n)`
- Total: `O(n)`

Space Complexity:

- No additional data structures are used, so the space complexity is **O(1)**.

Code

```
class Solution {
  public:
    void rotateArr(vector<int>& arr, int d) {
        d = d % arr.size();
        reverse(arr.begin(), arr.begin() + d);
        reverse(arr.begin() + d, arr.end());
        reverse(arr.begin(), arr.end());
};
```