**Problem solving :-**

**Sorting techniques :-**

**Selection sort**

class Solution {

    public int[] selectionSort(int[] nums) {

        int n = nums.length;

        for (int i = 0; i < n - 1; i++) {

            // Assume the current index is the minimum

            int minIndex = i;

            // Find the index of the actual minimum in the remaining array

            for (int j = i + 1; j < n; j++) {

                if (nums[j] < nums[minIndex]) {

                    minIndex = j;

                }

            }

            // Swap the found minimum with the current element

            int temp = nums[i];

            nums[i] = nums[minIndex];

            nums[minIndex] = temp;

        }

        return nums;

    }

}

**Bubble sort :-**

class Solution {

    public int[] bubbleSort(int[] nums) {

        int n = nums.length;

        for (int i = 0; i < n - 1; i++) {

            for (int j = 0; j < n - 1 - i; j++) {

                if (nums[j] > nums[j + 1]) {

                    // Swap adjacent elements

                    int temp = nums[j];

                    nums[j] = nums[j + 1];

                    nums[j + 1] = temp;

                }

            }

        }

        return nums;

    }

}

…

Rotating array by one postion left

class Solution {

    public void rotateArrayByOne(int[] nums) {

        int n=nums.length;

        int[] temp=new int[nums.length];

        for(int i=1;i<n;i++){

            temp[(i-1)%n]=nums[i];

        }

        temp[n-1]=nums[0];

        for(int i=0;i<n;i++){

            nums[i]=temp[i];

        }

    }

}

Rotating an array by k postion to right

class Solution {

    public void rotate(int[] nums, int k) {

        int n=nums.length;

        int[] temp=new int[n];

       for(int i=0;i<n;i++){

        temp[(i+k)%n]=nums[i];

       }

       for(int i=0;i<n;i++){

         nums[i]=temp[i];

       }

    }

}