Array 50 Practice Questions

Basic Level (1-20)

1. **Find the maximum element in an array.**
   - Input: `[1, 2, 3, 4, 5]`
   - Output: `5`
2. **Find the minimum element in an array.**
   - Input: `[4, 2, 7, 1, 9]`
   - Output: `1`
3. **Reverse the elements of an array.**
   - Input: `[1, 2, 3, 4, 5]`
   - Output: `[5, 4, 3, 2, 1]`
4. **Find the sum of all elements in an array.**
   - Input: `[1, 2, 3, 4, 5]`
   - Output: `15`
5. **Count the number of even and odd elements in an array.**
   - Input: `[1, 2, 3, 4, 5]`
   - Output: `Even: 2, Odd: 3`
6. **Print the elements of an array in alternate positions.**
   - Input: `[1, 2, 3, 4, 5, 6]`
   - Output: `[1, 3, 5]`
7. **Find the second largest element in an array.**
   - Input: `[12, 35, 1, 10, 34, 1]`
   - Output: `34`
8. **Find the second smallest element in an array.**

- Input: `[12, 13, 11, 15, 14]`
- Output: `12`

9. **Merge two sorted arrays.**
   - Input: `[1, 3, 5]` and `[2, 4, 6]`
   - Output: `[1, 2, 3, 4, 5, 6]`
10. **Check if an array is sorted.**
    - Input: `[1, 2, 3, 4, 5]`
    - Output: `True`
11. **Find the largest sum contiguous subarray (Kadane's Algorithm).**
    - Input: `[-2, -3, 4, -1, -2, 1, 5, -3]`
    - Output: `7`
12. **Left rotate an array by one position.**
    - Input: `[1, 2, 3, 4, 5]`
    - Output: `[2, 3, 4, 5, 1]`
13. **Left rotate an array by `k` positions.**
    - Input: `[1, 2, 3, 4, 5]`, `k=2`
    - Output: `[3, 4, 5, 1, 2]`
14. **Right rotate an array by one position.**
    - Input: `[1, 2, 3, 4, 5]`
    - Output: `[5, 1, 2, 3, 4]`
15. **Find the frequency of each element in an array.**
    - Input: `[1, 2, 2, 3, 3, 3]`
    - Output: `{1: 1, 2: 2, 3: 3}`
16. **Move all zeros to the end of an array.**
    - Input: `[0, 1, 0, 3, 12]`
    - Output: `[1, 3, 12, 0, 0]`

17. **Find the intersection of two arrays.**
    - Input: `[1, 2, 2, 1]`, `[2, 2]`
    - Output: `[2, 2]`
18. **Find the union of two arrays.**
    - Input: `[1, 2, 2, 1]`, `[2, 3]`
    - Output: `[1, 2, 3]`
19. **Remove duplicates from an array.**
    - Input: `[1, 2, 2, 3, 4, 4, 5]`
    - Output: `[1, 2, 3, 4, 5]`
20. **Find the element that appears only once in an array where all others appear twice.**
    - Input: `[2, 3, 5, 4, 5, 3, 4]`
    - Output: `2`

Intermediate Level (21-40)

1. **Find the missing number in an array of size `n` containing elements from `1` to `n+1`.**
    - Input: `[1, 2, 4, 6, 3, 7, 8]`
    - Output: `5`
2. **Find the duplicate number in an array of `n+1` integers where each integer is between `1` and `n`.**
    - Input: `[1, 3, 4, 2, 2]`
    - Output: `2`
3. **Rearrange an array so that `arr[i]` becomes `arr[arr[i]]`.**
    - Input: `[4, 0, 2, 1, 3]`
    - Output: `[3, 4, 2, 0, 1]`
4. **Find all pairs in an array that sum to a given value `x`.**

- Input: `[1, 5, 7, -1]`, `x=6`
- Output: `[(1, 5), (7, -1)]`

5. **Find the maximum product of two integers in an array.**
   - Input: `[1, 20, -1, -30]`
   - Output: `600`
6. **Implement a function to perform a binary search on a sorted array.**
   - Input: `[1, 2, 3, 4, 5]`, `key=3`
   - Output: `2` (index)
7. **Sort an array of `0s`, `1s`, and `2s` without using extra space (Dutch National Flag problem).**
   - Input: `[0, 1, 2, 1, 0, 2, 0, 1]`
   - Output: `[0, 0, 0, 1, 1, 1, 2, 2]`
8. **Find the common elements in three sorted arrays.**
   - Input: `[1, 5, 10]`, `[2, 3, 5]`, `[5, 6, 7]`
   - Output: `[5]`
9. **Rotate a square matrix 90 degrees clockwise.**
   - Input:

```
Copy code
1 2 3
4 5 6
7 8 9
```

   - Output:

```
Copy code
7 4 1
8 5 2
```

```
9 6 3
```

10. **Find the longest consecutive sequence in an array.**
    - Input: `[100, 4, 200, 1, 3, 2]`
    - Output: `4 (sequence: 1, 2, 3, 4)`
11. **Find the `k` th largest element in an array.**
    - Input: `[3, 2, 1, 5, 6, 4]`, `k=2`
    - Output: `5`
12. **Find the `k` th smallest element in an array.**
    - Input: `[7, 10, 4, 3, 20, 15]`, `k=3`
    - Output: `7`
13. **Rearrange the array in alternating positive and negative items.**
    - Input: `[1, 2, 3, -4, -1, 4]`
    - Output: `[1, -4, 2, -1, 3, 4]`
14. **Find the subarray with a given sum.**
    - Input: `[1, 4, 20, 3, 10, 5]`, `sum=33`
    - Output: `[20, 3, 10]`
15. **Find the median of two sorted arrays of equal size.**
    - Input: `[1, 3, 8, 9, 15]`, `[7, 11, 19, 21, 18]`
    - Output: `11`
16. **Sort an array based on frequency of elements.**
    - Input: `[4, 5, 6, 5, 4, 3]`
    - Output: `[4, 4, 5, 5, 6, 3]`
17. **Count pairs in an array with a given difference.**
    - Input: `[1, 5, 3, 4, 2]`, `diff=3`

- Output: `2 (pairs: (1,4), (2,5))`

18. **Find if there is a subarray with 0 sum.**
    - Input: `[4, 2, -3, 1, 6]`
    - Output: `Yes (subarray: [2, -3, 1])`
19. **Implement an algorithm to find the majority element.**
    - Input: `[3, 3, 4, 2, 4, 4, 2, 4, 4]`
    - Output: `4`
20. **Sort an array of strings based on length.**
    - Input: `["apple", "banana", "kiwi", "cherry"]`
    - Output: `["kiwi", "apple", "cherry", "banana"]`

Hard Level (41-50)

1. **Find the maximum length of subarray having equal number of 0s and 1s.**
    - Input: `[0, 0, 1, 0, 1, 1]`
    - Output: `4`
2. **Find the triplet that sum to a given value.**
    - Input: `[12, 3, 4, 1, 6, 9]`, `sum=24`
    - Output: `(12, 3, 9)`
3. **Find the minimum number of swaps required to sort the array.**
    - Input: `[4, 3, 2, 1]`
    - Output: `2`
4. **Maximum product subarray.**
    - Input: `[6, -3, -10, 0, 2]`
    - Output: `180`
5. **Given an array of `n` elements, find the maximum `j - i` such that `arr[j] > arr[i]`.**

- Input: `[34, 8, 10, 3, 2, 80, 30, 33, 1]`
- Output: `6`

6. **Find the smallest subarray with sum greater than a given value.**
   - Input: `[1, 4, 45, 6, 10, 19]`, `sum=51`
   - Output: `3 (subarray: [4, 45, 6])`
7. **Implement a program to merge `k` sorted arrays.**
   - Input: `[[1, 3, 5], [2, 4, 6], [0, 9, 10, 11]]`
   - Output: `[0, 1, 2, 3, 4, 5, 6, 9, 10, 11]`
8. **Find the maximum of all subarrays of size `k`.**
   - Input: `[1, 3, 1, 2, 0, 5]`, `k=3`
   - Output: `[3, 3, 2, 5]`
9. **Print all subarrays with 0 sum.**
   - Input: `[6, 3, -1, -3, 4, -2, 2, 4, 6, -12, -7]`
   - Output: `Multiple subarrays`
10. **Count the number of subarrays with a sum equal to `k`.**
    - Input: `[10, 2, -2, -20, 10]`, `sum=-10`
    - Output: `3`