🧠✨ Lecture 08 — PROBLEM SOLVING ON ARRAY & VECTOR (C++)

> “DSA is not about syntax.
> It’s about **how data flows through logic**.”

🔴 ① ARRAY KO FUNCTION ME PASS KARNA (MOST IMPORTANT CONCEPT)

❓ Problem

Array ko function me pass karte waqt:

- copy banti hai ❌
- ya original modify hota hai? ✅

🧠 Truth (First Principle)

👉 **Array by default reference ke jaise pass hota hai**

📌 Code (Tumhara hi code)

```
void printvalue(int a[], int n) {
    for(int i = 0; i < n; i++)
        cout << a[i] << endl;

    a[0] = 90;   // modification
}

int arr[5] = {10, 20, 11, 18, 30};
printvalue(arr, 5);
cout << arr[0];   // 90
```

🔍 Why?

- `arr` → base address pass hota hai
- Function original memory pe kaam karta hai

📌 Interview Line

> “Arrays decay into pointers when passed to functions.”

🔁 DEEP COPY vs SHALLOW COPY (ARRAY COPY)

❓ Problem

Agar ek array ko dusre me copy karna ho?

🧠 Solution

Loop ke through **deep copy**

```
for(int i = 0; i < 5; i++)
    arr2[i] = arr[i];
```

📌 Industry Insight

- C++ me raw arrays = manual copy
- `vector` me copy automatic & safe

🔎 ② LINEAR SEARCH (GFG + LEETCODE)

❓ Problem

Array me element exist karta hai ya nahi?

🧠 Logic

- Har element ko ek-ek karke check
- Match mila → index return
- Nahi mila → -1

⏱ Complexity

- Time: **O(n)**
- Space: **O(1)**

📌 GFG Version

```
int search(int arr[], int n, int x) {
    for(int i = 0; i < n; i++)
        if(arr[i] == x)
            return i;
    return -1;
}
```

📌 LeetCode (Vector)

```
int search(vector<int>& arr, int key) {
    for(int i = 0; i < arr.size(); i++)
        if(arr[i] == key)
            return i;
    return -1;
}
```

📌 Use Case

- Unsorted array
- Small data
- Brute force acceptable

🔄 ③ REVERSE ARRAY — TWO POINTER TECHNIQUE ⭐⭐⭐

❓ Problem

Array ko **in-place** reverse karna

🧠 Core Idea

- Left pointer → start
- Right pointer → end
- Swap & move inward

📌 Code (Exact)

```
void reverseArray(vector<int>& arr) {
    int n = arr.size();
    for(int i = 0; i < n / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[n - 1 - i];
        arr[n - 1 - i] = temp;
    }
}
```

🔍 Formula yaad rakho

```
Right index = (n - 1) - i
```

⏱ Complexity

- Time: **O(n)**
- Space: **O(1)**

📌 Same logic → **LeetCode 344 (Reverse String)**

🧠 ④ VECTOR — ARRAY KA INDUSTRY VERSION

❓ Array already hai, vector kyun?

Real life example:

- Phone contacts
- Orders list
- Notifications

👉 Size runtime pe fix nahi hota

🧠 Vector kya solve karta hai

Problem	Vector Solution
Fixed size	Dynamic size
Manual copy	Automatic
Unsafe	Safe
No methods	Rich API

📌 Basic Operations

```
vector<int> arr;
arr.push_back(90);
arr.pop_back();
arr.size();
```

📌 Industry Rule

> "Use array only when size is fixed & known at compile time."

📈 ⑤ CHECK SORTED & MONOTONIC ARRAY

🔷 Sorted (GFG)

```
bool isSorted(int arr[], int n) {
    for(int i = 0; i < n - 1; i++)
        if(arr[i] > arr[i + 1])
            return false;
    return true;
}
```

🔷 Monotonic (LeetCode 896)

```
bool isMonotonic(vector<int>& nums) {
    bool inc = true, dec = true;

    for(int i = 1; i < nums.size(); i++) {
        if(nums[i] < nums[i - 1]) inc = false;
        if(nums[i] > nums[i - 1]) dec = false;
    }
    return inc || dec;
}
```

📌 Difference

- Sorted → only increasing
- Monotonic → increasing OR decreasing

❓ ⑥ MISSING NUMBER IN ARRAY (VERY IMPORTANT)

🔷 Approach 1: Brute Force ❌

- $O(n^2)$
- TLE

🔷 Approach 2: SUM FORMULA ✅

```
missing = n*(n+1)/2 - actualSum
```

⏱ O(n), O(1)

🔷 Approach 3: XOR (INTERVIEW FAVORITE ⭐)

```
ans = XOR of array
x   = XOR of 1..n
missing = ans ^ x
```

📌 Why XOR works?

- x ^ x = 0
- Missing element bacha rehta hai

🧠 WHAT LECTURE 08 ACTUALLY TAUGHT YOU

Concept	Real Skill
Array in function	Memory understanding
Linear search	Brute force thinking
Reverse	Two pointer mastery
Vector	Industry readiness
Sorted/Monotonic	Pattern detection
Missing number	Math + bit manipulation

🏁 FINAL SUMMARY (SAVE THIS)

- Array function me **reference jaisa behave** karta hai
- Linear search → unsorted data
- Reverse → **two pointer = symmetry**
- Vector → dynamic, safe, industry standard
- Sorted & monotonic → comparison logic
- Missing number → **XOR = interview gold**

🧠🔥 INTERVIEW GOLD LINES

- “Array parameters decay into pointers.”
- “Two pointer reduces space to O(1).”
- “Vector abstracts memory management.”

✨ Made By @Harshal Chauhan