**Scenario: Debugging Bubble Sort Logic in C++**

**Scenario:**  
Investigated and corrected a logic bug in a Bubble Sort implementation. Although the program compiled and executed, it occasionally produced out-of-bounds access or incorrect results. Used GDB to trace execution and identify the faulty loop bounds.

**Problem Description:**  
The C++ program attempted to sort an array using Bubble Sort but produced incorrect results:

* Inner loop iterated over the entire array instead of stopping before the sorted portion, causing potential out-of-bounds access (arr[j+1]).
* The sorting result was sometimes incorrect or caused runtime warnings/errors.

This bug illustrates a common off-by-one error in nested loops, which can lead to both logical errors and memory safety issues in production code.

**Tools & Languages:**

* Language: C++
* Debugging Tool: GDB

**Initial Code (Buggy Excerpt):**

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

for (int j = 0; j < n; j++) { // <-- bug: should be j < n-i-1

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

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

**Issues Detected:**

* Inner loop iterates beyond the unsorted portion, causing **out-of-bounds array access** when j+1 == n.
* Sorting could fail for larger arrays due to invalid memory access.
* Illustrates a classic off-by-one error in nested loops.

**Fixed Code (Corrected Excerpt):**

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

for (int j = 0; j < n-i-1; j++) { // fixed: stop at n-i-1

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

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

**Corrections Made:**

* Updated inner loop condition to j < n-i-1 to avoid accessing arr[j+1] out-of-bounds.
* Ensured the algorithm only iterates over the unsorted portion of the array in each pass.

**Outcome:**

* Array is correctly sorted:

Original array: 5 2 4 1 3

Sorted array: 1 2 3 4 5

* GDB tracing confirmed no out-of-bounds access.
* Demonstrates the importance of careful loop bounds in sorting algorithms.

**Lessons Learned:**

* Small loop bound errors can cause major logical or memory issues.
* Always verify the iteration limits in nested loops.
* Debugging with GDB helps catch runtime errors and validate algorithm correctness.
* Even simple algorithms like Bubble Sort require careful attention to indexing.