**1. Using clone() (Shallow Copy)**

int[] backup = original.clone();

✅ **Pros:**

* Simple and efficient for **1D arrays** of primitive types (int[]).
* Creates a **new array** with copied values.

⚠️ **Cons:**

* For **multi-dimensional arrays (int[][])**, clone() only makes a shallow copy (copies references, not actual data).

**2. Using Arrays.copyOf()**

int[] backup = Arrays.copyOf(original, original.length);

✅ **Pros:**

* Creates a **new array** and **copies all elements** from original.
* Works well for **truncating** or **expanding** arrays (e.g., Arrays.copyOf(original, n) can adjust size).

⚠️ **Cons:**

* Only works for **1D arrays** (for 2D arrays, you'd need to copy each row separately).

**3. Using System.arraycopy()**

int[] backup = new int[original.length];

System.arraycopy(original, 0, backup, 0, original.length);

✅ **Pros:**

* **Fast and efficient** for large arrays (uses a native method).
* Works for **subarray copying** (System.arraycopy(original, startIndex, backup, destIndex, length)).

⚠️ **Cons:**

* **More verbose** than clone() or Arrays.copyOf().
* Still a **shallow copy** for multi-dimensional arrays.

**Comparison Table**

| **Method** | **Works for int[]?** | **Works for int[][] (deep copy)?** | **Creates a new array?** | **Performance** |
| --- | --- | --- | --- | --- |
| clone() | ✅ Yes | ❌ No (shallow copy) | ✅ Yes | 🔹 Fast |
| Arrays.copyOf() | ✅ Yes | ❌ No (shallow copy) | ✅ Yes | 🔹 Fast |
| System.arraycopy() | ✅ Yes | ❌ No (shallow copy) | ✅ Yes (must allocate manually) | 🔥 Very fast |

For **deep copying** a **multi-dimensional array**, you’d need to loop through rows and copy them separately:

int[][] backup = new int[original.length][];

for (int i = 0; i < original.length; i++) {

backup[i] = original[i].clone(); // Deep copy each row

}

**Final Recommendation**

For **1D int[] arrays**, prefer:  
✅ original.clone() for simplicity.  
✅ Arrays.copyOf(original, original.length) for flexibility.  
✅ System.arraycopy(original, 0, backup, 0, original.length) for performance.