

**Comparison with Other Data Structures**

1. **ArrayList**: Supports random access because it uses a contiguous array internally.
2. **LinkedList**: Does **not** support random access. Accessing an element requires traversing the list from the beginning or end (O(n) time complexity).
3. **Example Difference**:
   * arrayList.get(5) → Instant access.
   * linkedList.get(5) → Requires traversal from the head or tail to the 5th node.

ArrayList<String> list = new ArrayList<>();

list.add("A"); // Index 0

list.add("B"); // Index 1

list.add("C"); // Index 2

// Random access

String element = list.get(1); // Directly retrieves "B" without iteration

System.out.println(element); // Output: B

**Benefits of ArrayList**

* Dynamic resizing eliminates the need to manually manage array size.
* Provides random access and is more efficient for read operations compared to LinkedList.

**7. Drawbacks**

* Adding/removing elements at arbitrary indices is slower due to shifting.
* Frequent resizing can impact performance if the size grows unpredictably.

**Resizing Mechanism**

* Resizing occurs during addition if the current capacity is insufficient.
* Steps:
  1. A new array with a larger size is created.
  2. The contents of the old array are copied into the new one using Arrays.copyOf.
  3. The reference to the old array is replaced with the new array