As we mentioned in the previous article, an array has a fixed capacity and we need to specify the size of the array when we initialize it. Sometimes this will be somewhat inconvenient and wasteful.

Therefore, most programming languages offer built-in dynamic array which is still a random access list data structure but with variable size. For example, we have vector in C++ and ArrayList in Java.

*Operations in Dynamic Array*

Let's take a look at the usage of the dynamic array:

|  |
| --- |
| // "static void main" must be defined in a public class.  public class Main {  public static void main(String[] args) {  // 1. initialize  List<Integer> v0 = new ArrayList<>();  List<Integer> v1; // v1 == null  // 2. cast an array to a vector  Integer[] a = {0, 1, 2, 3, 4};  v1 = new ArrayList<>(Arrays.asList(a));  // 3. make a copy  List<Integer> v2 = v1; // another reference to v1  List<Integer> v3 = new ArrayList<>(v1); // make an actual copy of v1  // 3. get length  System.out.println("The size of v1 is: " + v1.size());  // 4. access element  System.out.println("The first element in v1 is: " + v1.get(0));  // 5. iterate the vector  System.out.print("[Version 1] The contents of v1 are:");  for (int i = 0; i < v1.size(); ++i) {  System.out.print(" " + v1.get(i));  }  System.out.println();  System.out.print("[Version 2] The contents of v1 are:");  for (int item : v1) {  System.out.print(" " + item);  }  System.out.println();  // 6. modify element  v2.set(0, 5); // modify v2 will actually modify v1  System.out.println("The first element in v1 is: " + v1.get(0));  v3.set(0, -1);  System.out.println("The first element in v1 is: " + v1.get(0));  // 7. sort  Collections.sort(v1);  // 8. add new element at the end of the vector  v1.add(-1);  v1.add(1, 6);  // 9. delete the last element  v1.remove(v1.size() - 1);  }  } |