

Lesson 8: Removing Elements

The Social Media Dilemma

You are writing an algorithm for your favorite social media app. This algorithm uses an array to store the names of every user someone is following. We know that people sometimes “break up” with friends for whatever reason. Therefore, the user needs the ability to unfollow or remove a friend from their list.

```
String unfollowString = "User to Remove";
for (int i = 0; i < followerList.length; i++){
    if (followerList[i].equals(unfollowString)){
        followerList[i] = null;           // No easy way to resize the now-smaller array
    }
}
```

How is the `remove()` method different from how we would remove items from an array?

Similar	Different
<ul style="list-style-type: none"> It is possible to remove elements from both types of data structures We can use for loops to traverse arrays and ArrayLists. 	<ul style="list-style-type: none"> Arrays are immutable in length, and must be recreated in order to lower the size. ArrayLists shift elements down in index to account for a removed element, dynamically resizing the ArrayList.

Removing with an Enhanced for Loop

Enhanced for Loop: Arrays

```
for(int price : prices) {  
    System.out.println(price);  
}
```

Enhanced for Loop: ArrayList

```
for(Integer price : prices) {  
    System.out.println(price);  
}
```

The enhanced for loop used with an array uses a primitive data type (int) whereas the enhanced for loop with an ArrayList uses an object (Integer).

WORD BANK

ConcurrentModificationException enhanced loop ArrayList array

Using an **enhanced for loop** while adding or removing elements from an ArrayList may cause a **ConcurrentModificationException**.

Arrays are immutable, which means the size cannot change.

Using `remove()` with an **ArrayList** completely removes the element, shifting the rest of the items.

Reflection

Question of the Day: How is removing data from an `ArrayList` different from removing data from an array?

removing items from an array doesn't change the size of the array, while removing elements from an `ArrayList` resizes the list

How does the idea of removing elements connect with what you already know?

We have already learned ways to remove elements with arrays, but with `ArrayLists` we must use methods to perform these operations rather than relying on manually removing elements within an array by referring to their index.