

Array Processing and Examples – Part II

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To program common array operations (§7.2.6) – several building blocks for problem solving using arrays
 - finding the minimum and maximum elements, and
 - random shuffling, and shifting elements.
- To simplify programming using the foreach loops (§7.2.7)

Processing Arrays

- Discussed the following building blocks
 - Initializing arrays with input values
 - Initializing arrays with random values
 - Printing arrays
 - Summing all elements, computing average
 - Counting the elements
- Now, discuss
 - Finding the largest element, finding the smallest element
 - Finding the index of the largest element, finding the index of the smallest element
 - Random shuffling
 - Shifting elements
- Enhanced for loop

Finding the value of the largest element

```
double max = myList[0];  
for (int i = 1; i < myList.length; i++) {  
    if (myList[i] > max) max = myList[i];  
}
```

// any logical errors that might occur?

Finding the value of the smallest element

```
double min = myList[0];
```

```
for (int i = 1; i < myList.length; i++) {
```

```
    if (myList[i] < min) min = myList[i];
```

```
}
```

// any logical errors that might occur?

Finding the index of the largest element

```
double maxIndex = 0;
```

```
for (int i = 1; i < myList.length; i++) {
```

```
    if (myList[i] > myList[maxIndex]) maxIndex = i;
```

```
}
```

Finding the index of the smallest element

```
double minIndex = 0;
```

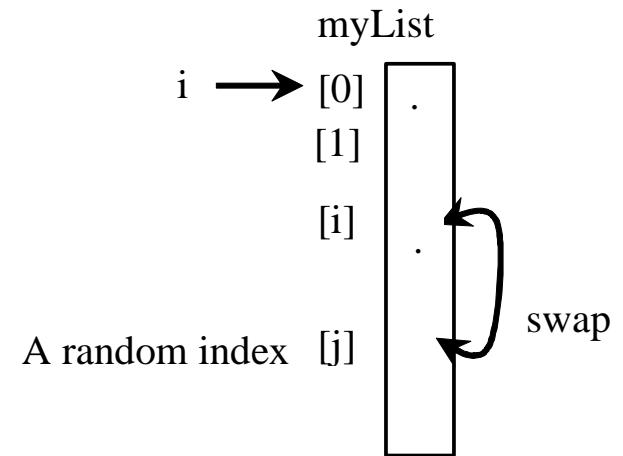
```
for (int i = 1; i < myList.length; i++) {
```

```
    if (myList[i] < myList[minIndex]) minIndex = i;
```

```
}
```

Random shuffling – 1 of 2

```
for (int i = 0; i < myList.length - 1; i++) {  
    // Generate an index j randomly  
    int j = (int)(Math.random() * myList.length);  
  
    // Swap myList[i] with myList[j]  
    double temp = myList[i];  
    myList[i] = myList[j];  
    myList[j] = temp;  
}
```



Random shuffling – 2 of 2

```
// any difference when compared to the previous solution
for (int i = 0; i < myList.length - 1; i++) {
    // Generate an index j randomly
    int j = i + (int)(Math.random() * (myList.length - i));

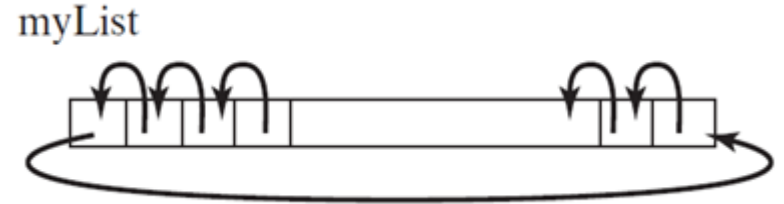
    // Swap myList[i] with myList[j]
    double temp = myList[i];
    myList[i] = myList[j];
    myList[j] = temp;
}
```

Shifting Elements

```
// retain the first element  
double temp = myList[0];
```

```
// shift elements left  
for (int=1; i<myList.length; i++) {  
    myList[i-1] = myList[i];  
}
```

```
// move the first element to fill in the last position  
myList[myList.length - 1] = temp;
```



Questions?

- Finding the largest element, finding the smallest element
- Finding the index of the largest element, finding the index of the smallest element
- Random shuffling
- Shifting elements

Enhanced For Loop

- JDK 1.5 introduced a new for loop that enables you to traverse the complete array sequentially without using an index variable. For example, the following code displays all elements in the array myList:

```
for (double value: myList) {  
    System.out.println(value);  
}
```

- In general, the syntax is

```
for (elementType value: arrayRefVar) {  
    // Process the value  
}
```

- You still have to use an index variable if you wish to traverse the array in a different order or change the elements in the array.

Questions?

- How many building blocks/examples can be rewritten using the enhanced for loop?