Lesson 8: More Array Algorithms

45 minutes

Overview

How do I plan and implement algorithms to find the minimum and maximum values in a 1D array?

Students explore the `File` class and use the `Scanner` class to read data from a file into a 1D array. In the process, students revisit the `ArrayIndexOutOfBoundsException` to identify debugging strategies to resolve off-by-one errors. Using data from a file, students develop algorithms to find the minimum and maximum values in a 1D array. Students plan their algorithms to identify the steps to solve the problem and write pseudocode to outline these steps.

Standards

Full Course Alignment

CSA Conceptual Framework

▶ **CON-2** - Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values

▶ **VAR-2** - To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure wihtout creating individual variables for each value.

Agenda

Warm Up (5 minutes)
- **Arrays and Loops**

Activity (35 minutes)
- **Working with Files**
- **Accessing Pairs**
- **Minimum and Maximum Values**

Wrap Up (5 minutes)
- **Glows, Grows, Want-to-Knows**
- **Assessment: Check for Understanding**
- **AP Classroom Topic Questions**

Objectives

Students will be able to:

- Create a one-dimensional (1D) array from a text file
- Identify debugging strategies to resolve off-by-one errors
- Implement an algorithm to determine the minimum and maximum value in a one-dimensional (1D) array

Preparation

- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **U3L8 Extra Practice** - Handout

Vocabulary

- **Text File** - a file that contains letters, numbers, and/or symbols but has no special formatting
- **off-by-one error** - an error that occurs when we try to access an index value that is greater than the length of the array

Teaching Guide

Warm Up (5 minutes)

Arrays and Loops

Remarks

> `while` loops and `for` loops are helpful for repeating a task and for traversing a 1D array.

Discuss: *How could we achieve this using a* `while` *loop? How could we achieve this using a* `for` *loop?*

Discussion Goal: With a `while` loop, the `Painter` object can be moved from (2, 5) to (2, 1) by moving forward until it is on a paint bucket. This condition can be checked using the `isOnPaintBucket()` method. With a `for` loop, the `Painter` object can be moved from (2, 5) to (2, 1) by moving four times.

Activity (35 minutes)

Working with Files (15 minutes)

Remarks

> As software engineers work with larger amounts of data, they often need to store the information in other files to read into their programs.

Do This: Review the lesson objectives.

Do This: Direct students to Level 1 on Code Studio to predict the program's outcome, then run the program to compare their predictions to the actual outcome.

1 **Predict: Reading a File**

Discuss: Click through the animated slide to display the prompts.

- *What do you notice about the code in this program?*
- *What do you wonder about the code in this program?*

Discussion Goal: Students notice that values are stored in a text file and read into the program. Students also notice that the values are stored in a 1D array. Students may wonder how they might work with larger files or with multiple types of values or how they could write data to a file.

Do This: Click through the animated slide to introduce the `File` class.

Teaching Tip

When introducing the `File` class, focus on the `File(String pathname)` constructor. In Java Lab, students can create a `File` object by specifying the name of the text file as a `String`.

Do This: Define *text file*.

Teaching Tip

Ask students if they have used a text editor such as Notepad. Have students identify the similarities and differences between the files created in these text editors and programs like Microsoft Word or Google Docs.

Discuss: Use the Hold That Thought strategy to discuss the prompt.

- *How can reading a file be useful in a program?*

Discussion Goal: Students note that a program can read data in a file. Students may also suggest that it makes the program more user-friendly and adaptable. The user does not need to enter the files directly into the array, and one program can process multiple data sets.

Discuss: Click through the animated slide to display the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What do you notice about the packages that are imported? Why do you think we need these?*
- *What do you think the instance variables represent?*

Discussion Goal: Students identify the packages that are imported and predict that these are needed to provide access to the `Scanner`, `File`, and `FileNotFoundException` classes. Students note that the instance variables represent a `Scanner` object to read the file and a `File` object for the file.

Discuss: Click through the animated slide to display the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What do you notice about the constructor?*
- *What do you think is the purpose of the* `setFile()` *method?*

Discussion Goal: Students notice that the constructor calls the `setFile()` method. Students realize that the `setFile()` method instantiates the `File` and `Scanner` object. Students note that this method is a mutator method so it can be used to change the file and update the `Scanner` object to read a different file.

Discuss: Click through the animated slide to display the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What do you think is the purpose of the* `createScanner()` *method?*
- *What do you notice about the code in this method?*

Discussion Goal: Students note that the method returns a `Scanner` object to read the file. Students notice that the code uses a `try/catch` block that instantiates the `Scanner` object or notifies the user that the file cannot be found if a `FileNotFoundException` occurs.

Discuss: Click through the animated slide to display the prompts. Use the Retrieve-Pair-Share strategy to discuss the prompts.

- *What do you think is the purpose of the* `getIntData()` *method?*
- *How is this similar to how we have read input from a user? How is it different?*

Discussion Goal: Students note that the method returns a 1D array of int values. Students realize that the `Scanner` object is used to check if there is a value to be read and reads the value into the array if a value is found. Students notice that the contents of the file are read using the same methods that they used to read user input, but that it first checks if there is data to read.

Teaching Tip

Students are not expected to know how to write code to read files on their own. Any activities in the curriculum that use file input provide the code for reading a file to students. The purpose here is to expose students to the functionality of the code so they understand how to use it when it is provided to them in upcoming activities.

Do This: Click through the animated slide to introduce the `Scanner` class methods for checking if there is input.

Accessing Pairs (10 minutes)

Remarks

> Files makes it easier for programs to apply the same algorithms to multiple data sets. This makes the program more flexible, but it also makes it more user-friendly because the user can just provide a different file to process and analyze using the same program.

Group: Place students in pairs.

Do This: Direct students to Level 2 on Code Studio to investigate the program with a partner. Students make the changes to the program as prompted.

2 **Investigate: Accessing Pairs**

Discuss: Click through the animated slide to display the prompts.

- *What is causing this error? How do we fix it?*
- *What are strategies we can use when debugging errors like this?*

Discussion Goal: Students note that the program had an `ArrayIndexOutOfBoundsException` because the array traversal looped one too many times. Students share that changing the condition resolved the error and allowed the program to function as expected.

Remarks

> When we are using `for` or `while` loops, we need to be sure that these iteration statements do not loop one time too many or one time too few. Doing so results in an off-by-one error.

Do This: Explain off-by-one errors.

Minimum and Maximum Values (10 minutes)

Remarks

> We have written algorithms throughout the unit to determine if elements in an array met specific criteria or find the sum or average of the values. We can also write algorithms to find the minimum and maximum values in an array.

Group: Place students in pairs.

Teaching Tip

These can be the same pairs as the previous activity.

Do This: Have students write pseudocode to find the minimum and maximum values of a 1D array.

Do This: Direct students to Level 3 on Code Studio to implement their algorithms.

3 **Finding Minimum and Maximum Values**

Wrap Up (5 minutes)

Glows, Grows, Want-to-Knows

Remarks

> We have learned a lot in this unit about storing values in 1D arrays and writing algorithms to find information.

Discuss: Click through the animated slide to display the prompts.

- *What was awesome about writing your code?*
- *What is one action you can take to improve your code?*
- *What questions do you have about today?*

Discussion Goal: Students share aspects of their program they enjoyed or strengths of writing algorithms and identify areas of improvement to strengthen their programming skills. Students also share any questions or misconceptions they may have.

Teaching Tip

Glows, Grows, Want-to-Knows is a strategy to help students identify their strengths and areas of improvement and share any questions or misconceptions.

Do This: Review the concepts covered in this lesson.

Display: Key Vocabulary

Assessment: Check for Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.

4 **Check for Understanding**

AP Classroom Topic Questions

To assign questions from the AP Classroom Question Bank that align with this lesson, create a custom quiz in AP Classroom by searching the Question Bank for the Essential Knowledge statements listed at the top of this lesson plan. You can find instructions and video demonstrations to do this on **AP Central**.

The following Topic Questions in AP Classroom can be assigned as a formative assessment for this lesson:

- Topic Questions 6.4

Note: *Some Learning Objectives and Essential Knowledge statements in the suggested Topic Questions are covered in later units.*

This curriculum is available under a
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes **contact us**.