Unit 5 - Lesson 7
2D Array Algorithms

Computer Science A

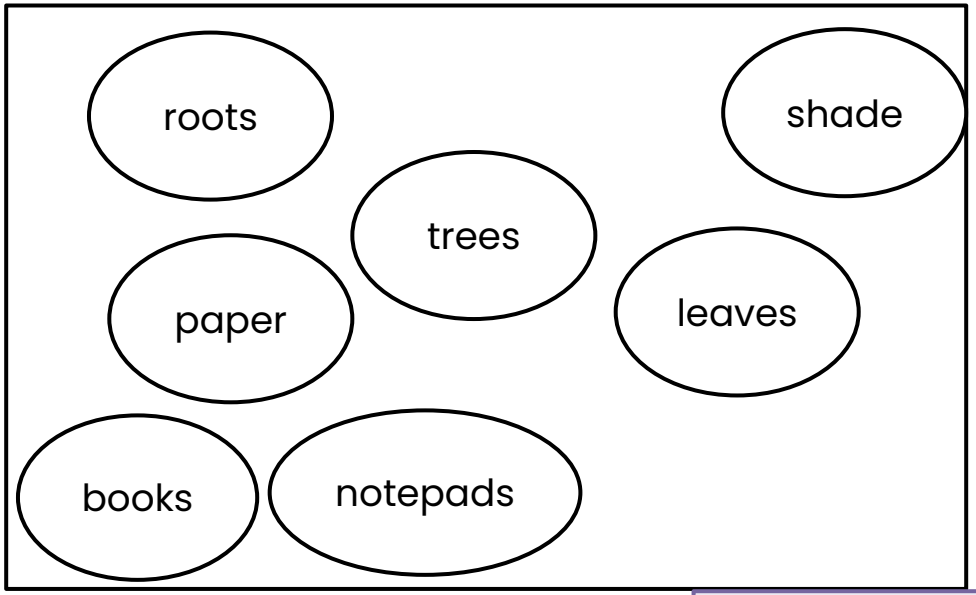
Warm Up

What do 2D arrays allow us to do that 1D arrays didn't?

✅ Do This:

Create a **concept map** by **brainstorming** any **concepts** and **ideas** that come to mind.

Example: What is a tree?



📝 **Unit 5 Guide**

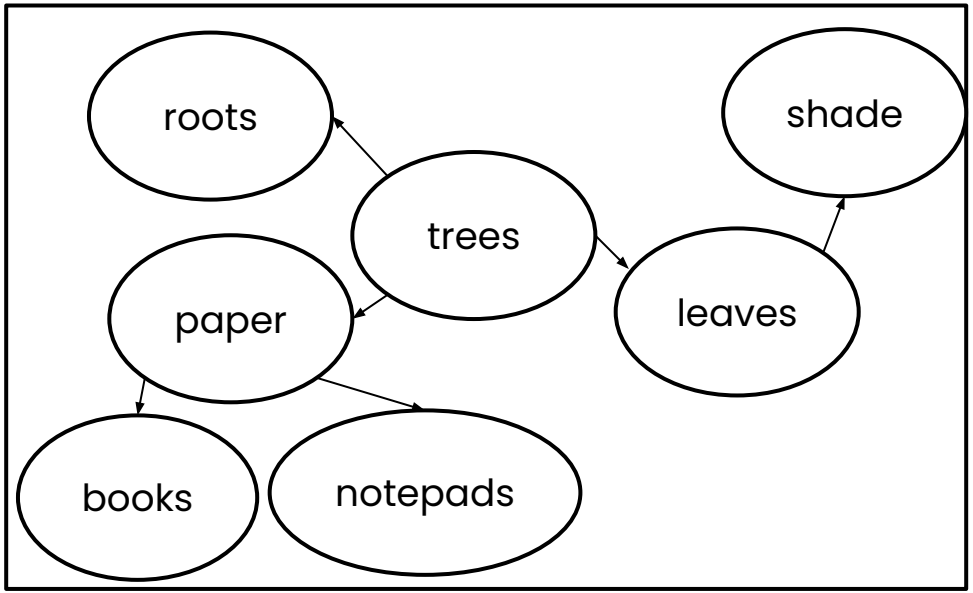
How are these concepts connected?

✅ Do This:

Review the **concepts** and **ideas** you came up with.

Draw **arrows** to **connect** the concepts.

Example: What is a tree?

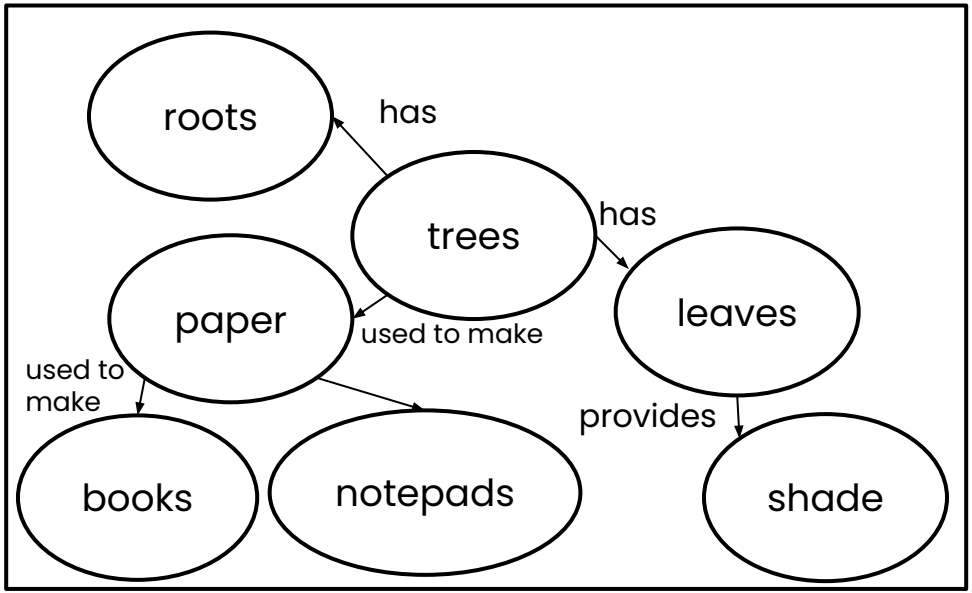


How are these concepts connected?

✅ Do This:

Label the arrows with **verbs** or **short descriptions** to identify the **relationships** between the concepts and ideas.

Example: What is a tree?



✅ Do This:

Share your concept map with a neighbor.

Compare the concepts and relationships you wrote.

Add to or **revise** your concept map based on your discussion.

Activity

🎯 Lesson Objectives

By the end of this lesson, you will be able to . . .

- Use the **`indexOf()`** method in the **`String`** class to determine if a specific character or **`String`** appears in a **`String`** object
- Implement an algorithm to search or modify elements in a two-dimensional (2D) array

🚀 Question of the Day

How can I use what I know about object-oriented programming and 2D arrays to plan and implement algorithms?

Predict and Run

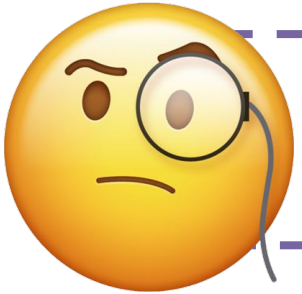
Navigate to Lesson 7, Level 1

✅ Do This:

1. Predict the output of the program

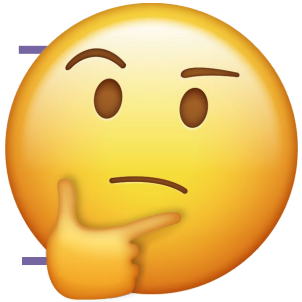
   There are no wrong answers!

2. Run it to compare your prediction with the results

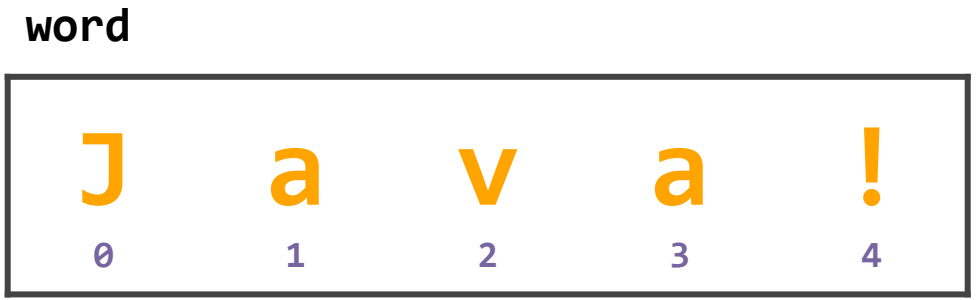


What did you notice about the code in this program?

What do you wonder about the code in this program?



Each character in a **`String`** is at an index which starts at **`0`**.



Investigate and Modify

Navigate to Lesson 7, Level 2

✅ Do This:

1. Investigate the code on **Levels 2 through 4**
2. Make changes as prompted and observe the results

What did you discover from the modifications you made to the code?

The **`indexOf()`** method returns the index in the **`String`** of the first occurrence of **`str`**.

```
String message = "Hello World!";
System.out.println(message.indexOf("o");
System.out.println(message.indexOf("World");
System.out.println(message.indexOf(" ");
```

```
4
6
5
> _
```

Self Check

Consider the following code segment.

```
String text = "The quick brown fox jumps over the lazy dog.";
int result = text.indexOf("fox");
```

What will be the value of result after executing this code segment?

A 3

B 16

C 19

D -1

🎉 Skill Building

Navigate to Lesson 7, Level 5

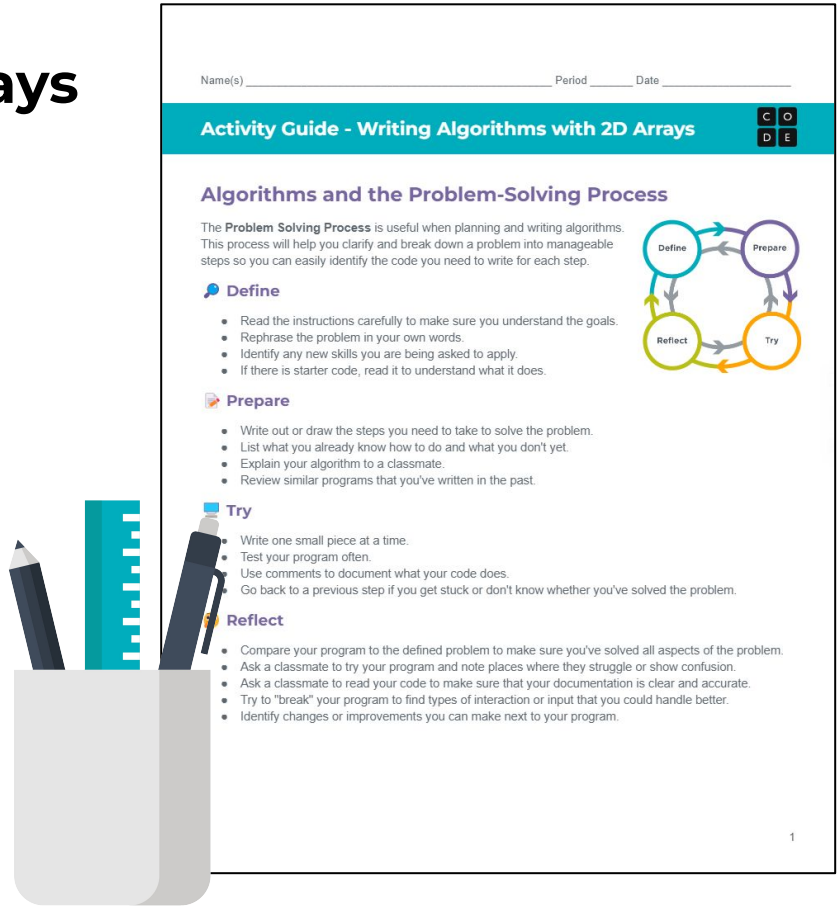
✅ Do This:

Level 5 - Practice using the `indexOf()` method to find the location of a character or `String` in a `String` object

Writing Algorithms with 2D Arrays

📍 You and your partner should have:

- Writing Algorithms with 2D Arrays activity guide
- Planning Algorithms Manipulatives
- pen / pencil



Name(s) _____ Period _____ Date _____

Activity Guide - Writing Algorithms with 2D Arrays

Algorithms and the Problem-Solving Process

The **Problem Solving Process** is useful when planning and writing algorithms. This process will help you clarify and break down a problem into manageable steps so you can easily identify the code you need to write for each step.

Define

- Read the instructions carefully to make sure you understand the goals.
- Rephrase the problem in your own words.
- Identify any new skills you are being asked to apply.
- If there is starter code, read it to understand what it does.

Prepare

- Write out or draw the steps you need to take to solve the problem.
- List what you already know how to do and what you don't yet.
- Explain your algorithm to a classmate.
- Review similar programs that you've written in the past.

Try

- Write one small piece at a time.
- Test your program often.
- Use comments to document what your code does.
- Go back to a previous step if you get stuck or don't know whether you've solved the problem.

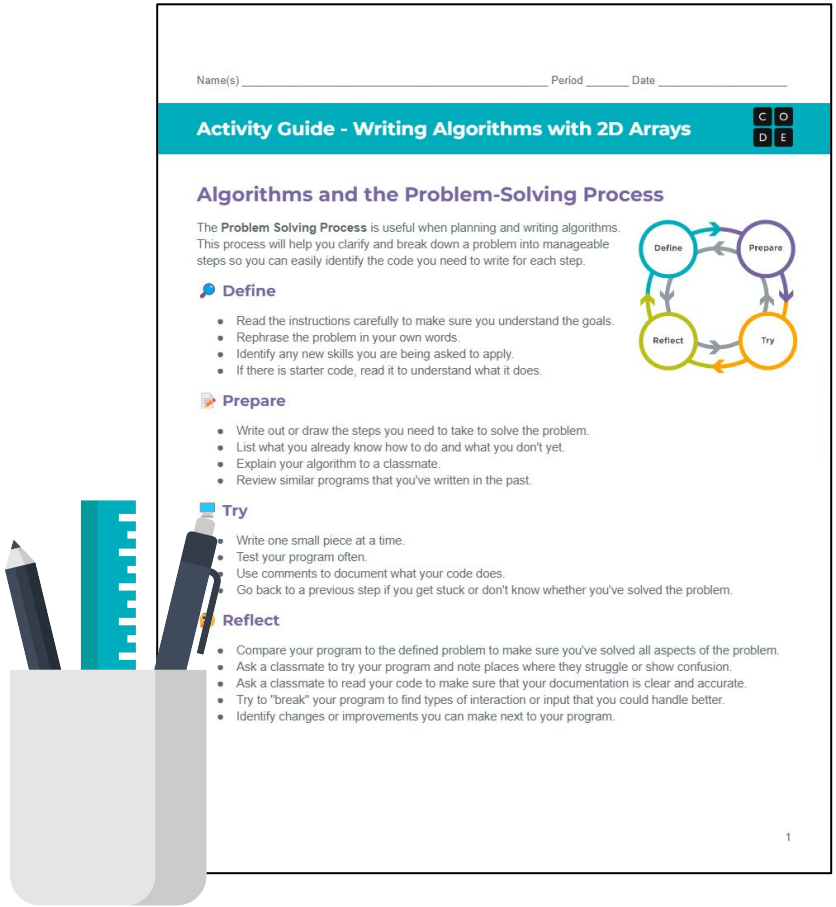
Reflect

- Compare your program to the defined problem to make sure you've solved all aspects of the problem.
- Ask a classmate to try your program and note places where they struggle or show confusion.
- Ask a classmate to read your code to make sure that your documentation is clear and accurate.
- Try to "break" your program to find types of interaction or input that you could handle better.
- Identify changes or improvements you can make next to your program.

1

✅ Do This:

Work with your partner to write pseudocode for one of the problems.



Name(s) _____ Period _____ Date _____

Activity Guide - Writing Algorithms with 2D Arrays

Algorithms and the Problem-Solving Process

The **Problem Solving Process** is useful when planning and writing algorithms. This process will help you clarify and break down a problem into manageable steps so you can easily identify the code you need to write for each step.

Define

- Read the instructions carefully to make sure you understand the goals.
- Rephrase the problem in your own words.
- Identify any new skills you are being asked to apply.
- If there is starter code, read it to understand what it does.

Prepare

- Write out or draw the steps you need to take to solve the problem.
- List what you already know how to do and what you don't yet.
- Explain your algorithm to a classmate.
- Review similar programs that you've written in the past.

Try

- Write one small piece at a time.
- Test your program often.
- Use comments to document what your code does.
- Go back to a previous step if you get stuck or don't know whether you've solved the problem.

Reflect

- Compare your program to the defined problem to make sure you've solved all aspects of the problem.
- Ask a classmate to try your program and note places where they struggle or show confusion.
- Ask a classmate to read your code to make sure that your documentation is clear and accurate.
- Try to "break" your program to find types of interaction or input that you could handle better.
- Identify changes or improvements you can make next to your program.

1

Practice

Navigate to Lesson 7, Level 6

Do This:

1. **Level 6 -** Check for Understanding
2. **Level 7 -** Implement your algorithm for your problem

It's time to . . .

MY CODE!

T — **Tell** them something you like about their code.

A — **Ask** them something about the code.

G — **Give** a suggestion for improvement.

Wrap Up

Closing the Loop

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?

🎉 Today, you learned about . . .

- Use the **`indexOf()`** method in the **`String`** class to determine if a specific character or **`String`** appears in a **`String`** object
- Implement an algorithm to search or modify elements in a two-dimensional (2D) array

🚀 Question of the Day

How can I use what I know about object-oriented programming and 2D arrays to plan and implement algorithms?