

Solving Problems

Your Tasks (Mark these off as you go)

- ☐ Complete the problems
- ☐ Pair up and reflect
- ☐ Compare algorithms
- ☐ Wrap up
- ☐ Receive credit for this lab guide

☐ Complete the prompts

Today we are going to explore how computer scientists think about problem solving. An important skill we will explore will be recognizing patterns and similarities.

Review the problems below for one minute, and then move around the room and collect the information needed to solve the problems.		
	Prompt	Information
1	Find a person who has more programming experience than you	
2	Find a person who has less programming experience than you	
3	Find a person who has the same amount of programming experience as you	
4	Find a person who has taken AP Computer Science Principles from Ms. Pluska	
5	Find a person who has taken AP Computer Science Principles from someone other than Ms. Pluska	
5	Find a person who has more than two years of programming experience	
6	Find a person who has no programming experience	
7	Find the person with the most programming experience	
8	Find a person who has written programs in Python	
9	Find a person who has written programs in Java	
10	Find a person who has made money from a program they have written	

☐ Pair up and reflect

Locate the person you found for prompt 3 above. Write their name below. Indicate one fun fact about that person.	
Person	
Fun fact	

Discuss the following prompts with your group. Write your responses below.

How did you go about solving each of the problems?

For which problems did you need to do something similar to solve them? For example problems 1 & 2 are very similar.

□ Compare algorithms

We just thought about whether problems are similar. Now we're going to look at whether we're solving the same problem.

Consider the algorithms below. Use the grids provided to trace the algorithms.

Algorithm 1

```
MOVE_FORWARD()
TURN_RIGHT()
MOVE_FORWARD()
TURN_RIGHT()
MOVE_FORWARD()
TURN_RIGHT()
MOVE_FORWARD()
TURN_RIGHT()
```

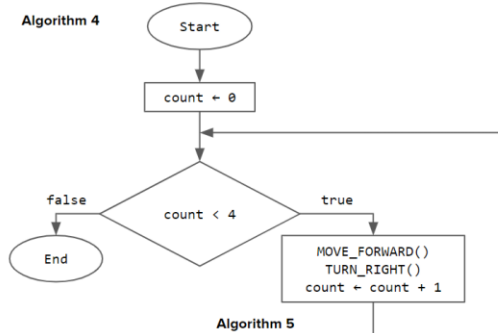
Algorithm 2

```
REPEAT 2 TIMES
{
  MOVE_FORWARD()
  MOVE_FORWARD()
  TURN_RIGHT()
  MOVE_FORWARD()
  TURN_RIGHT()
}
```

Algorithm 3

```
moves = ["F", "R", "F", "R", "F", "R", "F", "R"]
FOR EACH move IN moves
{
  IF (move = "F")
  {
    MOVE_FORWARD()
  }
  ELSE
  {
    TURN_RIGHT()
  }
}
```

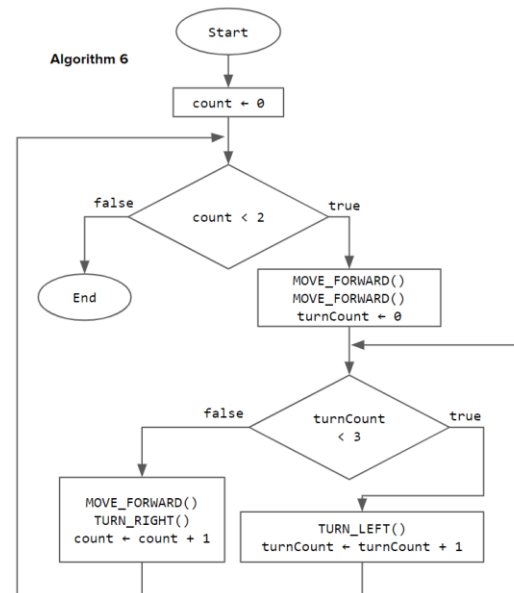
Algorithm 4



Algorithm 5

```
REPEAT 2 TIMES
{
  REPEAT 2 TIMES
  {
    MOVE_FORWARD()
  }
  REPEAT 3 TIMES
  {
    TURN_LEFT()
  }
  MOVE_FORWARD()
  REPEAT 3 TIMES
  {
    TURN_LEFT()
  }
}
```

Algorithm 6



With your partner decide which of these programs are “the same” as one another.

Discuss with your group which of these algorithms are like one another. How did you decide that?

- **Wrap-up**

In the first part of this activity, you had to solve a problem. In the second part of this activity, you explored algorithms. In the space below write a definition for each of these terms.

Problem

Algorithm

Pick a person in your group to share out the following prompts you answered above.

- Which problems in the first part of this activity were similar? Explain.
- Which algorithms were similar? Explain.
- What is a problem?
- What is an algorithm?
- What is a fun fact about yourself? What is fun fact about each person in your group?

☐ **Receive credit for this lab guide**

Submit this portion of the lab to Pluska to receive credit for the lab guide.