6

LESSON NAME:

Algorithms

Lesson time: 45–60 Minutes : Prep time: 10 25 Minutes (depending on whether you have tangrams available or need to cut them by hand)

Main Goal: Explain that the same thing can be accomplished many different ways, and sometimes there are "better" ways than others.

OVERVIEW:

This lesson covers algorithms. Using tangram shapes and graph paper, the first exercise will show how important it is to make each instruction as clear and unambiguous as possible. Afterward, the class will explore how many ways you can fold paper into a rectangle, noting how some methods can take more or fewer folds than others.

OBJECTIVE:

Students will —

- Practice creating algorithms that describe real-world directions
- Learn to think about solving a problem many different ways
- Think about creating more "efficient" solutions to problems

MATERIALS:

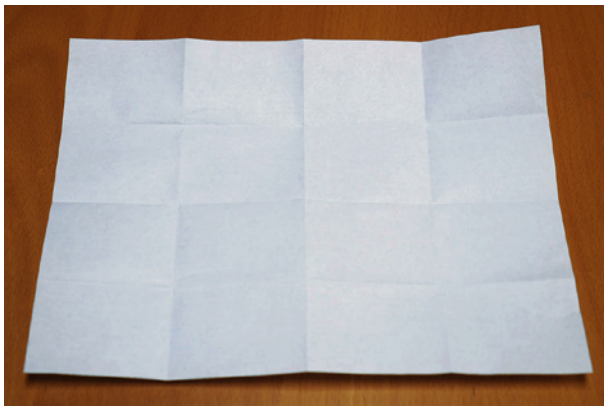
- Tangram Image Pack (one set per pair)
- Tangram Pieces (one set per pair)
- Graph paper (five or six sheets per pair)
- One piece of blank paper

PREPARATION:

Lay out one image pack, one set of tangrams, and a packet of graph paper for each group.

Set aside a blank piece of paper for each group.

Fold a sample sheet of paper to make 16 equal rectangles



VOCABULARY:

Algorithm—A list of steps that allow you to complete a task

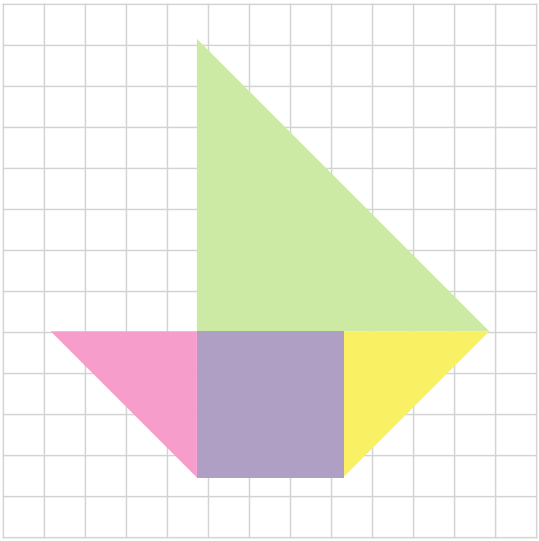
Ambiguous—Having more than one meaning

Efficiency—Having the best outcome for the least amount of work

Evaluate—To work at an answer

Hamburger Fold—This means to fold a paper in half the wide way

Hot Dog Fold—This means to fold a paper in half the long way



REVIEW:

This review segment is intended to get the class thinking back to the last lesson. If you are covering these activities out of order, please substitute your own review subjects here.

Class Participation Questions:

- What did we do in our last lesson?
- What did the set color block do?

Elbow Partner Discussion:

- How would we use what we learned last time to make a square with different colors on all sides?

INTRODUCE:

This lesson introduces some mind-blowing concepts. Not only can a computer "misunderstand" what you mean for it to do, but you can tell a computer to do the same thing several different ways. We'll walk you through two different activities that will help your students get experience with both cases.

Our first game is based on an old favorite called Tangrams. Tangrams are a Chinese geometric puzzle consisting of a square cut into seven pieces that can be arranged to make various other shapes. Our activity has been designed to use the same Tangram pieces as the classic puzzle, but with a twist. We won't require the use of every single piece each time, and we will be laying our shapes out on graph paper.

To begin this exercise, tell the class that computers "understand" things differently than we do. In large part, this is because computers can't "guess" what we want based on our tone of voice or our body language. If you tell your friend "Aperture is a hard word. Can you spell that?" It is very likely that your friend will try to spell "aperture." If your friend is a computer, however, it would probably spell the word "that." This is because a computer is going to take its instruction directly from the phrase it's given. If you give an **ambiguous** instruction, it will **evaluate** it the way it has been told to, whether it is what you meant or not.

Now it's time to break up into groups. We are going to see how hard it is to give clear instructions. One person in each group will be a "programmer" who will grab a picture that is made of a series of shapes. Another person will get a packet of shapes and a piece of paper and be the "computer." The two will sit back-to-back, and that's where the fun begins!

The programmer needs to try to describe their image to the computer, without ever letting the computer see it. Programmers can use whatever words or phrases they want to help their computer rebuild the original image, but they cannot use sound effects, or body movements.

(As the teacher, you can decide how you want to limit each turn. Number of instructions? Time in minutes? A combination of both? You can also choose whether or not to give programmers a second chance to communicate after they see how their first attempt turned out.)

Once the turn is over, the computer becomes the programmer, and someone else becomes the computer. How many rounds does it take before the computer succeeds in recreating the original image? What were the first mistakes? What were the most common ones? Which were the easiest to fix?

After you have come back together to discuss the successes and failures of the last activity, prepare the class to change gears.

Let them know that the last challenge had them perfecting algorithms for clarity, while this one will have them perfecting algorithms for **efficiency.**

Hold up your pre-folded piece of paper. Show them that there are 16 equal rectangles that were made only by folding the paper. Ask them how many of them think they can fold their paper to make the same rectangles. If it's more than half of the class, then you're probably okay to disperse right into the exercise. Otherwise, show them one of the most obvious ways to get the result (hamburger, hamburger, hamburger, hamburger). Decide based on your class whether you are going to have them work together or individually on the rest of the task.

Giving each group (or person) a sheet of paper, assign them one mission at a time.

1) Can you fold 16 equal rectangles into a sheet of paper?

2) Can you find a second way to do it?

3) Can you find a third way to do it?

At this point, you may want to suggest they start keeping track of the order of their folds on another piece of paper.

4) How many ways can you find to fold the exact same rectangles?

5) How many folds does it take to get to that result?

6) What is the highest number of folds that you can make to create those rectangles?

7) What is the smallest number of folds that you can make to create those rectangles? (4 folds)

Isn't it interesting that you can get the exact same result so many different ways, and some of those ways take so many more folds than others? What if we had only arrived at the solution that took six folds, and we had two million pieces of paper to create rectangles out of? That's four million extra folds that we would have made that we didn't have to! Not very efficient.

The idea of efficiency is very important in computer science, because computers run about 113 million instructions per second. If your program has more instructions than it needs, then you are actually adding *time* to how long it takes a program to run. Can you imagine adding *days* to how long it takes to load a web page? If you don't think about efficiency at all, you could do exactly that!

Sometimes, it helps to write a program that works first, then cut out any steps that were unnecessary (remember the graph paper drawings?) Other times, you learn tricks that help you keep your program efficient from the very beginning. In the case of our paper folding, the trick is to stick to folding exactly in half each time; that way we double the amount of creases that each fold makes every time. In computer science, the idea of cutting a problem in half shows up over and over again, so keep that trick in mind as you come across more and more difficult problems in the future!

ADJUSTMENTS:

K-2: It may be best to do all of this as a class, bringing two students to the front to do the programmer/computer exercise so that everyone can learn from previous mistakes. The folding paper exercise may work best if the instructor does all the folding and has the class count. If the class tends to stay on task, it can be helpful for them all to have their own sheet of paper in case they want to try to think ahead.

3-5: Small groups are the key here. The class may be tempted to spend nearly all of their time on the programmer/computer exercise. If this happens, feel free to guide the paper-folding activity (as above) rather than having the students do it on their own.

6-8: This group may actually get bored with the programmer/computer exercise if left too long, as they will probably zero in on perfection rather quickly. It may help to give them extra obstacles when describing their picture, such as not being able to say the name of the final image.

Older kids may prefer to do the paper folding activity in pairs. If you challenge them to just figure out how many ways they can fold the 16 rectangles, you can save all of the other questions for after they're done with the hands-on work. This provides some great reflective "ah-ha" moments.