Algorithms and Artwork – Aaron Oppenheimer

Purpose

A key component of any science or math class is the instruction of methods used to complete a task or conduct an experiment. The methodologies we pass to our students are nothing more than heuristics that allow us to navigate a STEM-based perspective on the world. When these processes are broken down into basic steps, they become analogous to the algorithms used in computation to run complex simulations or tackle big-data problems. However, students are not made aware of this connection, between their processes and those involved in computation, despite the obvious similarities between the two. Algorithms and algorithmic thinking play an integral role in every aspect of our lives, but these concepts often go unnoticed because they are so entrenched in our daily routines. This lesson seeks to bring the methods behind algorithmic thinking to the forefront of thought to help students better grasp the difficulties associated with computational algorithm design and implementation in a very contextualized environment. Additionally, a fundamental understanding and utilization of algorithmic teaching transforms a student into an active learner, where each step is followed by the questions, "where am I now," "where do I want to be," and "what can I do next?"

Overview

This lesson is aimed at introducing the concept of algorithmic design such that the students can easily apply it to their daily routines and a method that is crucial to the course material. From a computational perspective, this application is referred to as an *abstraction,* in which the framework or rules of a process are applied to numerous contexts because at the base level, the goal is the same. *Abstraction* is a key cross-cutting concept in that it does very well to connect material from several STEM areas; for example, we may *abstract* the principle of conservation of momentum taught in physics to any number of applications, like conservation of mass in reactions, or conservation of energy throughout an ecosystem. In this way, *abstraction* becomes a necessary linker that helps tie the STEM fields together. To introduce and address *abstraction*, the lesson is split into three parts, initiated with a lecture concerning the ubiquitous nature of algorithms throughout our lives and introducing the context specific technique. Following this, the students are broken up into groups for an activity to showcase a key difficulty in algorithm design and implementation, namely translation and transformation of information, and lastly there is a wrap-up discussion for frustrations and applications of algorithmic thinking.

The details of the activity are relatively simple. Each group of students is given a set of instructions for various origami models. Each student selects a model, and attempts to successfully recreate the model from the visual instructions. In this way, the students generally experience the difficulty of interpreting simple steps in a foreign context. Upon completion, each student must then dictate instructions that guide another student to recreate his or her model, without showing the instructions or the finished product. Similarly, this generally poses a problem of how to state very basic steps in a clear enough manner that they can be executed by anyone. Finally, as a take-home assignment, it is suggested that the students attempt to formally write instructions that can be followed to recreate his or her model, without visual aid, to be completed by students as a follow-up assignment.

Reach for the Stars is a GK-12 program supported by the National Science Foundation under grant DGE-0948017. However, any opinions, findings, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the Foundation.

Student Outcomes

- Students will be able to describe familiar routines in the context of algorithms and iterations.
- Students will demonstrate understanding for the translational difficulties associated with computational algorithm design.
- Students will be able to successfully define an algorithm for (solving linear equations, balancing chemical reactions, deconstructing force diagrams, etc.) and employ said method to solve new problems.

Standards Addressed

This lesson addresses the following science and engineering practices and may be applied to a few core disciplinary concepts, depending on the method/technique taught in conjunction.

- Asking questions and defining problems – in algorithm design, it is crucial to outline the information that is necessary for a computer, or person, to complete a task. This means defining the problem and specifying the assumptions, boundaries, etc.
- Obtaining, evaluating, and communicating information – information translation and transformation is key in algorithm design and implementation. This is the crux of the activity portion of this lesson
- This lesson may pertain to those processes in evaluating chemical reactions (conservation of atoms/elements, solving Hess's law problems, etc.), in forces and motions problems (constructing and resolving multiple force vectors, conservation of momentum in multiple bodies, etc.), or in a plethora of mathematical techniques (long division, solving systems of linear equations, etc.).

Time

Lesson is intended to fit within a 45-50 minute period, but if taught in tandem with a new technique in class, the lesson is flexible to cover two regular periods.

Level

This lesson is appropriate for any age group, but should be taught in synch with a new method or technique (solving linear equations, balancing chemical equations, deconstructing force diagrams, etc.) to reinforce importance of algorithmic thinking within classroom context.

Materials and Tools

- Basic Origami Instructions (4 sets of three templates, such as those provided below)

group a	group b	group c	group d
cicada	panda	fish	bat
lion	Samurai hat	snake	T-Rex
penguin	Scottie dog	swan	tumble bird

- Origami Paper (suggested size 6" squares, ~5 sheets per student)
- Algorithms and Artwork Worksheet

Preparation

It is suggested the instructor have some sort of demonstrable, simple algorithm available as a tangible example (i.e. tie/bowtie, tied shoelace, Sudoku puzzle, etc.) for students to better grasp the concept.

Furthermore, it's helpful to set up the origami stations prior to class, so as to ease the transition between lecture and activity.

Prerequisites

If taught in conjunction with a new method, it may be preferable to introduce the method in a previous lecture and simply refer to it in a new sense, but the lesson is flexible in that you may simply introduce a new method as an algorithm and teach it in that style.

Background

There is no prerequisite knowledge necessary for the algorithm abstraction part of the lecture, and the part pertaining to the technique taught in conjunction will require context specific information. For example, if the method is balancing chemical reactions, it is necessary for students to be familiar with the concept of conservation of mass across a chemical reaction.

Teaching Notes

The activity associated with this lesson has quite the tendency to derail, so it's important to remind students frequently that the primary goal is to successfully teach another student how to fold a model without visual aids, just verbal instruction. The students should have fun, but not lose the message concerning the difficult nature of algorithm implementation.

Assessment

There are a few key points for assessment throughout the lesson. During the lecture, a formal dialogue with the students should be established, foremost to gauge what they're preconceived notions of "algorithm" are, and furthermore to dynamically gauge how well they're able to abstract the notion of algorithmic design onto everyday tasks and classroom methods. During the wrap-up, it's essential to pose the question, "what were the difficult parts of this task?" The ideal answers are "it's hard to understand simple steps in a language that we can't read," and "it's hard to tell someone how to do simple things without showing them explicitly." Lastly, the take-home assignment should be another good metric for how well students have grasped the idea of algorithmic design and implementation, based on how well their instructions can be followed to recreate a model. Additionally, a possible exit slip is as simple as name two examples of algorithms not discussed in class; one must be an algorithm learned in school, and the other must be from outside of school.

Additional Information

N/A