1 Introduction

In a point in history when students must be educated to solve problems that don’t yet exist, children must develop their minds’ ability to reason sequentially, logically, and clearly. While modern discussions of secondary level mathematics education in the United States tend to emphasize cultivating conceptual understanding in conjunction with procedural fluency, there is room to consider a student’s potential as a creator of algorithms.

This creative role is especially significant because of students’ likelihood to be placed in a position of responsibility with respect to new systems implemented in the future. In 2020, high school graduates will need to interact with complex systems in meaningful and concrete ways which require algorithmic fluency.

In Faster Isn’t Smarter (2009), past president of the National Council of Teacher of Mathematics (NCTM) Cathy Seeley discusses the power of patterns in the mathematics classroom. She says, “The patterns that underlie key mathematical concepts such as equivalence or proportionality help students make sense of otherwise disconnected bits of knowledge” (p. 122). She writes that “an algorithm reflects a fundamental generalization of a pattern,” and for maximum efficiency, students interlize “the smallest number of procedures that will serve them for the greatest number of situations” (p. 121).

2 Technology in the Classroom

In a position statement posted on their website, the NCTM speaks strongly in favor of the role of technology in mathematical pedagogy:

Technology is an essential tool for learning mathematics in the 21st century, and all schools must ensure that all their students have access to technology. Effective teachers maximize the potential of technology to develop students understanding, stimulate their interest, and increase their proficiency in mathematics. When technology is used strategically, it can provide access to mathematics for all students.1

The primary outlet for students in secondary level classrooms to make their algorithmic intuitions concrete is the common handheld graphing calculator. Texas Instruments’ TI series

1http://www.nctm.org/about/content.aspx?id=14233 (March 2008)
of graphing calculators has become ubiquitous in our nation’s classrooms over the past several decades. TI-BASIC, the language used in user-defined programs, provides a number of powerful features. Because of its indirect-memory referencing and conditional branching capabilities, TI-BASIC is a Turing complete formal computing system. That is to say, although the language itself is not particularly complicated, the capabilities at students’ disposable are, in a sense, computationally infinite.

3 Primary Goal

I’d like to create a series of calculator programming modules for use in secondary level mathematics classrooms. Students will develop skill in explicitly constructing algorithms while delving deeper into various mathematical topics. By the end of the series, students will have a firm grasp of the specific programming capabilities of common classroom electronics. They will be able to transfer this knowledge set into new domains that require creation of algorithms.

4 Computer Scientific Topics

- Cognitive development of algorithmic complexity
- Algorithmic embedding and encapsulation
- Recursion
- Solving games and classic problems, such as the Towers of Hanoi problem or Sudoku

5 Mathematical Content

Many mathematical phenomena offer entry points for the creative design of algorithms. As students investigate the patterns underlying the procedures required for an algorithm, they necessarily modularize a complex phenomenon. Students develop their skill in associating patterns with procedures generating them and assimilating these procedures into an algorithm shedding light on its mathematical underpinnings.

Potential topics include:

- Composition of functions
- Numerical approximation of derivatives and integrals
- Finding roots of polynomials
- Fractals
- Fermi problems
- Pascal’s Triangle
- Number theory
– Factoring and testing primality
– Divisibility in various bases

6 Deliverables

Pedagogical Tools I’ll provide a series of student-centered modules sequencing the discovery and implementation of various algorithms. I’d like to provide accompanying guides for educators.

Sample Programs I’ll include model programs for each module.

Final Paper I’ll write an essay describing what I set out to achieve and discussing how well I managed to achieve it. I’ll justify the pedagogical choices I made and connect the modules to the mathematical and computer science content I wanted them to deliver.