Overview: Number Sequences

Students are asked to implement one of three different algorithms. Then they are asked to do some critical thinking and find patterns in their output, or potentially a single (constant-time) formula to replace their (linear-time) algorithms.

Prerequisite Material

This activity requires knowledge of:

- ❑ Loops
- ❑ Conditionals
- ❑ Mathematical Operations

Solutions

- ❑ Algorithm 1 converges to the greatest common factor of the two inputs.
- ❑ Algorithm 2 generates a Fibonacci sequence based on the two inputs.
  - There exists a closed-form solution (constant-time formula) for Fibonacci sequence, but it's complex to derive.
  - It's probably important for students to acknowledge that it could exist, and then it can be shown that it does. See:
  - http://en.wikipedia.org/wiki/Fibonacci_number
- ❑ Algorithm 3 generates a number sequence that seems always to converge to 1 (Collatz's Conjecture). Nobody has yet proven this conjecture and may be unprovable.

Number Sequences

For this activity, divide into three groups, each with a laptop. Each group will be given a different algorithm by the instructor.

a. Implement the algorithm you have been given. Each algorithm takes one or two integers as input. These can be hard-coded, or read from the console with a Scanner. Each algorithm generates a number sequence, and these can be printed out to the console.

b. Once you have implemented the algorithm, run your program with a variety of different inputs. Think about the following questions:

1) What does your algorithm do? How does it behave?
2) How does the number sequence end? What is the meaning of the final number?
3) Does it always end the same way? Can you be sure?

Algorithm Cards

a. Number Sequence 1

This algorithm takes two integers as inputs, and generates a sequence of number pairs.

1. Begin with two positive integers as the first pair.
2. If both numbers are equal, STOP.
3. The next pair in the sequence is:
    a. The smaller member of the original pair
    b. The difference between the original pair
4. Go to step 2.

b. Number Sequence 2

This algorithm takes two integers as inputs, and generates a sequence of numbers.

1. Begin with two integers as the first two elements.
2. Let j be the length of the series we want to generate.
3. If j is zero, STOP.
4. The next number in the sequence is the sum of the two previous numbers.
5. Let j = j – 1.
6. Go to step 3.

c. Number Sequence 3

This algorithm generates a sequence of numbers.

1. Begin with any integer.
2. Repeat this step:
    a. If your number is even, divide it by 2 to find the next number.
    b. Otherwise, multiply it by 3 and add 1 to find the next number.

This algorithm doesn't have a "STOP" step. Start by running step two 1000 times, and see if the number sequence converges. If it does, stop when it reaches this number.