# Lesson 4 Algorithm correctness

## Learning objectives

1. Apply algorithmic approaches to problems in mathematics, e.g. solving a linear system.
2. State the definition of algorithm correctness: *that the algorithm is guaranteed to produce the desired output for all allowable inputs*.
3. Determine whether an algorithm is correct from its description or pseudocode.

## Materials

Magformers magnet set with booklet, small white board with tic tac toe game partially drawn, 2 towers-of-hanoi sets, 2 decks of cards

## Agenda

1. Solving a linear system algorithmically – 20 min
   1. Derivation of the formula for the point of intersection
   2. Python demo
2. Algorithm correctness – 20 min
   1. Dog feeding
   2. Python demo of an incorrect algorithm: guessing grade level based on age
3. Logic lab – 50 min
   1. Instructions and set up
   2. Do it!

Solving a linear system from a CS perspective

In Grade 10, we solved systems of equations like

**1) 4*x* – 3*y* = 25**

**2) 3*x* – 8*y* = -10**

You did it by hand in about 4 minutes. And you hated it, right?

What if we could teach a computer to do it? Then it could do the work for us!

But we want the computer not just to solve that one problem now on the board, but ***any*** problem of the form:

**1) Ax + By = C**

**2) Dx + Ey = F**

Let’s turn the method you learned into an algorithm!

Let’s solve the 1st problem on our own. Then we’ll develop the formula on the right.

|  |  |
| --- | --- |
| 1) 4x – 3y = 25 Times coeffs by 3  2) 3x - 8y = -10 Times coeffs by 4  1a) 12x – 9y = 75  2a) 12x -32y = -40 Subtract!  -9y – (-32y) = 75 – (-40)  23y = 115 Divide by 23  y = 115 / 23  y = **5** Sub into (1)  1) 4x – 3(5) = 25 Isolate 4x  4x = 25 + 15 Divide by 4  x = 40/4  x = **10**  So y = 5 and x = 10 | 1) Ax + By = C Times coeffs by D  2) Dx + Ey = F Times coeffs by A  1a) ADx + DBy = DC  2a) ADx + AEy = AF Subtract!  DBy - AEy = DC - AF  (DB – AE)y = DC – AF Divide by DB-AE  **y = (DC – AF) / (DB – AE)** Sub into (1)  1) Ax + B**y** = C Isolate Ax  Ax = C – By Divide by A  **x = (C – By) / A**  So y = (DC – AF) / (DB – AE) and x = (C-By)/A |

Demo the Python program *Solve by elimination.py*

### Algorithm correctness

What’s wrong with this robot algorithm for feeding the dog?

TURN SELF ON

GO TO GARAGE

FIND DOG FOOD BAG

OPEN BAG

SCOOP 250 GRAMS DOG FOOD

TURN SELF OFF

POUR FOOD INTO BOWL

What’s wrong with this algorithm for guessing the grade level of a child from its age?

Demo the (wrong) Python program *Faulty grade-level estimator.py*

and correct it with the students.

**Definition of algorithm correctness**

An algorithm is *correct* if it is guaranteed to generate the desired output   
for all allowable inputs.

Example 3: Picking a random winner from a group

“Suppose I want someone in the class to take down my attendance. What’s my algorithm for deciding?”

*Algorithm 1*

*Take attendance*

*Ask for a volunteer to take it down*

*Pick the first person to raise hand*

Is A1 correct? No, because what if no one raises their hand?

Thus, it’s not guaranteed to generate the desired output (i.e. a single winner)

*Algorithm 2*

*Take attendance*

*Run Hunger Games tournament*

*Winner takes down the attendance*

Is A2 correct? No, because what if there are two winners, like in the book?

Thus, A2 is not guaranteed to generate the desired output.

*Algorithm 3*

Take attendance

Eenie meenie miney moe.

Is A3 correct? Yes! EMMM always results in a single winner.

Does that make it the best algorithm? No, because there are other correct algorithms that run faster (i.e. pick someone at random)

Moral: As computer scientists, we try to design algorithms that are:

* Correct
* Efficient

## Finish HW #1-3