

Programming & Computational Thinking

Algorithms in computing and everyday life

Computational Thinking

Remember talking about computational thinking and these key principles?

- abstraction
- generalization
- composition & decomposition
- creativity
- data and information
- algorithms

Review – Abstraction

- What does it mean?
- How is it used in everyday examples?
- How about with computer or IT examples?

Review – Generalization

- What does it mean?
- How is it used in everyday examples?
- How about with computer or IT examples?

Review – Decomposition

- What does it mean?
- How is it used in everyday examples?
- How about with computer or IT examples?

Review – Composition

- What does it mean?
- How is it used in everyday examples?
- How about with computer or IT examples?

Algorithms

This week we'll talk about algorithms.

- What do you think of when you hear the word algorithm?
- What do you think it means?
- Where are algorithms used?

Definition

A finite set of unambiguous instructions that, given some set of initial conditions, can be performed in a prescribed sequence to achieve a certain goal and that has a recognizable set of end conditions.

~*American Heritage Dictionary*

Examples

A recipe is an example of an algorithm.

It is a set of steps that must be performed in sequence to achieve the end product.

What other everyday examples of an algorithm can you think of?

Multiple Choices

There are often many solutions to the same problem, so how do you choose?

Must consider the trade-offs and evaluate them for your specific situation.

Recipe Trade-offs

Multiple chocolate chip cookie recipes

- What things would you consider to choose?
- Are there trade-offs? Do you sacrifice one things for another?

Computer Algorithms

What trade-offs do you think we consider for computer algorithms?

Come up with two things. Focus on the connection between the two, more so than being “right”.

Watch out Google...

Search algorithms are commonly used in computing, and are easy to think about.

As we do this, be careful to pay attention to each step of the algorithm.

Don't skip anything! This is the hardest part.

Find a number

- Pick a number and write it on a sheet of paper.
- Split into 3 groups:
 - Back two rows are one group
 - First row is one group
 - Second row is another group
- Take everyone's sheet and think about how you would find one number out of all of them.
- Record the steps so that someone else can repeat it.
- Think like a computer - sequences, choices, loops.

Reflecting

- What was the gap between your largest and smallest number? Did that matter?
- How many numbers did you have overall? Did that matter?
- Did you have any duplicate numbers? Did that matter?
- Trade-offs?

Why is this important?

Take a few minutes to write down why you think this topic is relevant to you and your future career choices.

When you are done, share that with one or two of your neighbors.

Lab Intro

Explore how different algorithms perform

Today's lab is a little different - it is more about exploration of the ideas, than writing clever code.

Lab - Sorting

You'll work with a testing framework to explore how different sort algorithms work.

- Be sure to use the starter code.
- There are 3 sprites - one for each algorithm.
- Edit the green reporter block at the top only.
- Do not modify the testing code! (under the hat block)

Lab – Performance

The final part of lab categorizes algorithms by performance:

- constant time
- linear
- quadratic (n^2)
- logarithmic

This is more reading than coding, but you may be asked to discuss this on a test.