

Balance It Out

Teacher Materials

In this activity, students will guess which order the eight bottles of varying masses should be, and write an algorithm that will sort the bottles from least to greatest mass while only comparing two items at a time.

Acknowledgements and License

This activity was contributed by Keith Kramer, R M Marrs Middle School for UNO's TED 8006 course.

Lesson Details

Grade Level: 6-8th graders

Activity Length: 5-15 minutes

Format: Students working in pairs, materials for 1 pair

Connection to Standards

This activity contains content and practices addressing elements of the following CSTA K-12 Computer Science Standards (2017):

CSTA Std	Description
1A-AP-08	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
1A-AP-10	Develop programs with sequences and simple loops, to express ideas or address a problem.
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
1A-AP-12	Develop plans that describe a program's sequence of events, goals, and expected outcomes.
1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
3B-AP-10	Use and adapt classic algorithms to solve computational problems.

This activity contains content and practices addressing elements of the following Nebraska K-12 Technology Scope & Sequence Standards:

NE Tech Standard	Description
Computational Thinking 1	Create algorithms, or series of ordered steps, to solve problems.
Computational Thinking 2	Decompose a problem into smaller more manageable parts.
Computational Thinking 5	Optimize an algorithm for execution by a computer.
Computational Thinking 7	Evaluate algorithms by their efficiency, correctness, and clarity.

Bin Materials Checklist

#	Description
8	Containers of varying masses
2	Role Cards
1	Balance Scale
5	Notepads
10	Pencils

Lesson Vocabulary Words

algorithm an ordered sequence of instructions

Lesson Setup / Introduction

Students need to be paired up and assigned to be either the programmer or the computer. They should have all the materials from the bin. Student will then read their corresponding role cards for further instructions.

Connection to Computing

Sorting is a fundamental action for many applications and an area where computers excel. There are hundreds of different sorting algorithms, with new ones still being discovered. Because computers can do the same thing continually over and over again, at a phenomenal speed, they are excellent at sorting large numbers of items.

Sorting algorithms generally fall into two categories: comparison sorts and non-comparison sorts. Comparison sorts use comparison of values to accomplish the sorting. Non-comparison sorts use something other than the whole value. An example of a non-comparison sort is *radix sort* which compares only a single digit at a time.

Sorting algorithms are usually classified according to their complexity (how fast can they sort large numbers of items in the best, worst, and average case), their memory usage (how much memory do they use while sorting), and stability (do equal values remain in relative order after sorting).

This activity builds on everyday objects, weights. It simulates how a computer would do sorting in that the students can only compare two values at a time, just like a computer. There is a great deal of problem solving in this activity.

Further Related Resources

- There are many sorting visualizations and activities available on the internet. Here are a few:

- <https://brilliant.org/wiki/sorting-algorithms/#properties-of-sorting-algorithms>
- Sorting Out Sorting video <https://www.youtube.com/watch?v=SJwEwA5g0kM>
- 15 Sorting Algorithms in 6 Minutes
<https://www.youtube.com/watch?v=kPRAOW1kECg>
- Sorting Algorithms Animations
<https://www.toptal.com/developers/sorting-algorithms>
- People dancing to illustrate sorts https://www.youtube.com/playlist?list=PLuE79vNc5Wi6q34LsQcaJ7ISQ8u0yMaL_
- Having students sort themselves by age, height, etc. is a great interactive activity in class



Answer Key

Lightest → Heaviest

$B \rightarrow D \rightarrow E \rightarrow F \rightarrow H \rightarrow G \rightarrow A \rightarrow C$

Balance It Out

Student Instructions

Between you and your partner, choose one person to be the “Computer” and one to be the “Programmer”. Read the following description of your role or look at your Role Card.

Programmer

- The programmer writes an algorithm that will sort the containers by mass.
 - Algorithm only allows for checking the mass of two containers at a time.
 - Algorithm should “loop” until sorted.
 - The programmer cannot touch the containers, they instruct the computer how to compare and sort the containers.
 - For example: “Take two grocery containers, weigh them, place the lighter one to the left. Then. . .”

Computer

- The Computer must follow the programmer’s instruction exactly (even if the instructions won’t properly sort the items).
- Certain functions are assumed to be understood:
 - How to make sure the balance starts off zeroed
 - How to measure mass
 - Knowing your left and right

Warm up: Together, use the scale to order 3 bottles by weight from lightest to heaviest. What were your steps? How did you do it?

Now get another container and see if you can follow those same steps to put all four in order. Make sure to write down your steps - this is your algorithm.

Challenge: Now see if your algorithm will work for a larger set of weights.

- Mix up the eight bottles into random order.
- Each student or group writes an algorithm that will sort the bottles into a lightest to heaviest order.
- Algorithms may only compare two bottles at a time.
- After the algorithms have been written, take turns being the “Programmer” and the “Computer”. See the programmer and computer role cards for instructions.

- “Programmer” cannot touch the bottles. The programmer can only read the algorithm to the “Computer”.
- “Computer” must do exactly what the “Programmer” says.
- Record a tally sheet and write the total moves it took to order the bottles.

Now that you've tried your own sorting algorithm try the following algorithm together. Remember to keep track of the moves. This sorting algorithm is known as **Quick Sort**

- Choose a random bottle and place it on the right side of the balance
- Now take each of the remaining bottles, one at a time, and place it on the left side of the scale. If the bottle on the left is lighter than the right side, place it in a group on the table on the left side of the scale. If the bottle on the left is heavier than the right side, place it in a group on the table on the right side of the scale.
- Now there should be two groups (a lighter group on the left and a heavier group on the right). Place the bottle from the right side of the scale in the middle of the two groups.
- Now choose one of the new groups and repeat the process.
- Continue to repeat the above process until they are all sorted.
- Once this process is repeated until each bottle is in its own group, they will be in order from lightest to heaviest.

Was this algorithm better than yours? Why or why not?



Resupply

- Make sure there is enough paper