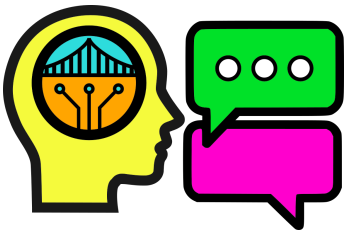


Sorting algorithms: unplugged

Lesson summary

Target Audience: *12-16 years old. Advanced level.*

Duration: *50 minutes*

Learning goals:
In this activity, students will learn how some common data sorting algorithms work through an unplugged and hands-on approach using a scale and several containers of different weights. The aim of this activity is to understand what happens when we ask to sort TV series on a streaming platform, photos in our gallery, or products in an online shop. In this case, we will always use the same criterion (the weight of each container), but that scale or criterion could be anything (the popularity of TV series, the time in the photos in the gallery, or the price in an online store).
Thanks to this activity, students will be able to:

- Sort any data set using a simple tool that only tells us which element comes before and which comes after when comparing two elements with each other.
- Find the largest or smallest element in a data set.
- Compare the advantages and disadvantages of some of the best-known sorting algorithms (selection sort, bubble sort, quick sort).

Online or offline: offline

Computational Thinking:

- **CT-foundations**:
    - Decomposition: following a ‘divide and conquer’ approach in recursive sorting algorithms such as quicksort.
    - Generalization: trying to use the same approach for any set of elements that has to be sorted.
    - Patterns: finding the relationships between the data and the possibility of sorting it by different criteria.
    - Algorithmic Thinking: systematically reproducing the steps of sorting algorithms.

Materials

- Salt / sand / rice (+ 1 spoon).
- 7 or more opaque containers.
- Rope / wool.
- Precision scale (only for the preparation, not needed later).
- Toy scale or clothes hanger.

Preparation

1. Fill the containers with different amounts of salt/sand/rice. Measure them with the precision scale in order to make them different.

2. Cut a small piece of rope/wool and close the container to make it easier to be hung in the clothes hanger.

3. Test the clothes hanger used as a scale:

Or use a toy scale (easier and funnier):

Lesson Description - Sorting algorithms

Introduction (5 minutes)
Welcome students and explain the challenge:

"Do you know what happens when you click the button to sort a set of products by price, novelty, or popularity in an online shop? The data is sorted using a sorting algorithm.

If you had 10 books on your table, what steps would you take to arrange them in a neat pile with the largest book at the bottom and the smallest at the top? Can you think of other ways to do this? Are some of them better or worse than others?"

Core

Warm-up Activity (10 minutes)

- Encourage students to think about the different criteria used to sort things: weight, price, volume, colour, etc.

  Questions for reflection:

  - Can we quantify any property of an object?
  - Can we sort what we cannot quantify?
  - Do we need to have an exact value for each object, or is it enough to be able to compare two objects with each other?

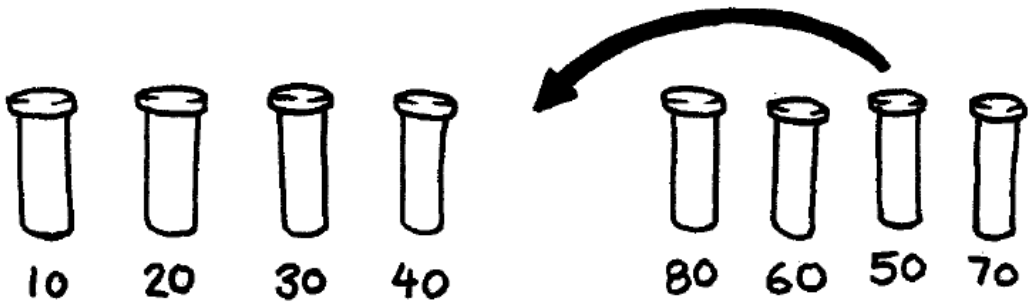
  Think of examples that support or contradict your answers.

- Connect the activity to Computational Thinking.

  - Algorithmic thinking: write down all the steps necessary to sort three objects according to their weight.
  - Generalisation: what would happen if there were five objects? What if there were 1,000? Would the same approach work?
  - Decomposition: can you think of a strategy to divide a big sorting problem (e.g., sorting 1000 cards) into smaller ones?

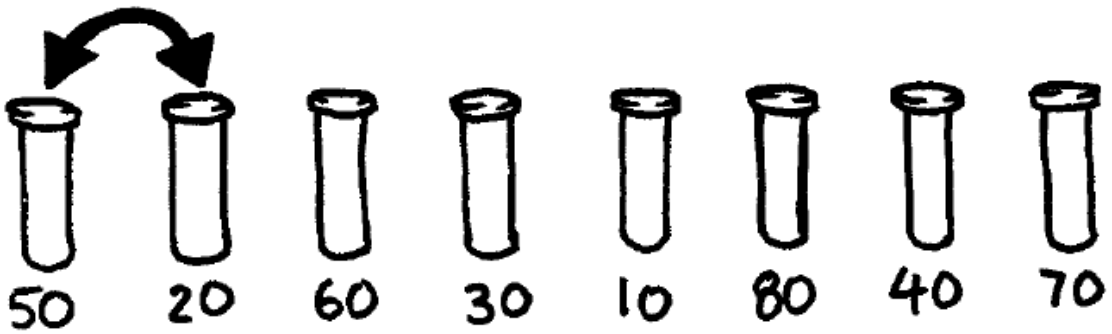
Main Activity – Sorting algorithms

Selection Sort (10 minutes)



- First, divide the students into groups (make as many groups as you have scales or clothes hangers).
- Second, provide each group with a balance or hanger and a set of containers of different weights.
- Third, have each group scramble their containers.
- Fourth, encourage each group to follow the selection sort algorithm:
    a. Find the lightest item in the set of items. How to find the lightest item:
        - Pick on an item as a candidate.
        - Compare it with the rest of the items.
        - Each time an item is lighter than the candidate, use it as the new candidate.
    b. Remove the lightest item from the set of items and place it on the left side of a new set of items.
    c. Repeat steps a and b until there are no items left.
    d. When this algorithm is complete, the items will be sorted in the new set.

Bubble Sort (10 minutes)

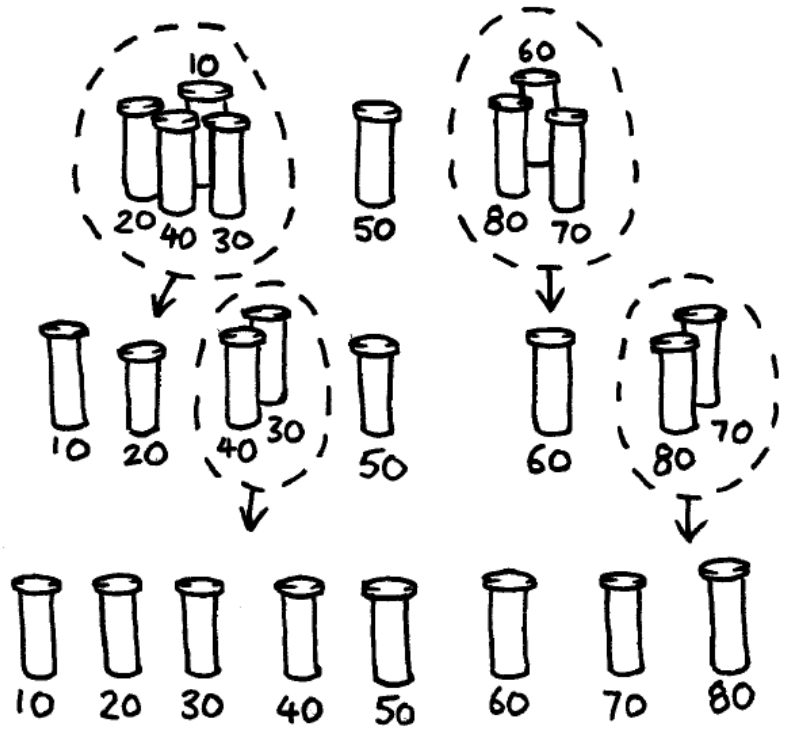


- Again, divide the students into groups, provide each group with a balance or hanger and a set of containers of different weights, and have each group scramble their containers.
- Then, encourage each group to follow the bubble sort algorithm:
    a. Compare each item with the next one and change their positions if they are not ordered.
    b. Repeat from the beginning until there is no change to be made.

Reflection / Discussion (10 minutes)

- Which is better, Select sort or Bubble sort? Why? Think about cases where one is faster than the other. Which of those cases is more likely to happen in real data?
- Can these algorithms take advantage of a “divide and conquer” approach? Can several people ordering the same set at the same time be faster using these algorithms?

Quick Sort (10 minutes)



- Again, divide the students into groups, provide each group with a balance or hanger and a set of containers of different weights, and have each group scramble their containers.
- Then, encourage each group to follow the quick sort algorithm:
    a. Pick a container randomly from the set, this will be the "pivot".
    b. Place all the containers lighter than the pivot on the left, and all the containers heavier than the pivot on the right.
    c. Repeat for the set of containers on the left and the set of containers on the right.
- Reflection: Can this algorithm take advantage of a “divide and conquer” approach? Can several people ordering the same set at the same time be faster using this algorithm?

Debrief and Evaluation (5 minutes)

- Ask reflection questions:
    - Would you be able to invent another sorting algorithm? Do you think your algorithm is better than the ones we have seen for any specific case?

- Would you be able to invent an algorithm to disorder any data set? Would they always be disordered in the same way? When do you think these types of algorithms are used?

Appendix 1

Comparison of several sorting algorithms

https://www.toptal.com/developers/sorting-algorithms