Finding the Right Balance: Facilitating the Exploration of Sorting Strategies using 3D-Printable Weights and Scales

Annika Vielsack
Department of Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
vielsack@kit.edu

Martina Landman*
Department of Computer Science and Technology
University of Cambridge
Cambridge, United Kingdom
ml2258@cam.ac.uk

Abstract

Sorting algorithms are a corner stone of algorithmic education. The computational problem of sorting objects is simple and accessible, with a wide variety of algorithms with different approaches and characteristics available. Introducing sorting algorithms in an explorative manner, however, faces a major challenge: students tend to sort objects intuitively and holistically, bypassing the step of comparing two objects. This renders algorithmic differences and advantages moot. In order to address this, we expand on an idea from the CS Unplugged programme and present a set of weights and balances we developed for hands-on exploration. The entire set is easily 3D-printable, making it scalable and thus possible to provide enough material for entire classes with no additional supplies needed.

CCS Concepts

• **Social and professional topics** → **K-12 education**; *Computational thinking*; • **Theory of computation** → **Sorting and searching**.

Keywords

Computer Science Education, Sorting Algorithms, Algorithmic Thinking, Computational Thinking

ACM Reference Format:
Annika Vielsack and Martina Landman. 2026. Finding the Right Balance: Facilitating the Exploration of Sorting Strategies using 3D-Printable Weights and Scales. In *28th Australasian Computing Education Conference (ACE 2026), February 09–13, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3786228.3786257

1 Introduction

The problem of Sorting in general and Sorting Algorithms in the specific is a major topic in Computer Science. Therefore, Sorting Algorithms are included in Computer Science Curricula worldwide [7], leading them to be taught in many introductory computer science courses. As a result, *sorting* has been part of computer science education for a while. Sorting is exemplary for a family of algorithms that solve the same problem and produce the same output but vary in their approach, their characteristics and their complexity. From an educational perspective this offers possibilities to transfer newly gained knowledge about one algorithm to other similar but distinct algorithms.

Unsurprisingly, a lot of teaching material about sorting algorithms exists. While activities focusing on coding have their own benefit, we focus on unplugged activities, which are learning interventions without using a computer or digital devices [12].

The advantage of using unplugged activities is the opportunity to introduce students to the algorithmic approach of sorting without prior programming, or other digital experience. Using physical materials and objects rather than, e.g., digital visualisations, provides an accessible approach regardless of the availability of digital learning resources [2]. Additionally, they offer a haptical explorative environment for experiencing the learning concepts.

Our aim is to provide teaching materials that are easy to obtain, low in cost, simple to use – for both students and teachers – and support students in exploring sorting algorithms intuitively and holistically. For this reason we developed the existing idea of sorting weights with a balance scale further. The presented sorting weights together with the matching scale can be used in versatile settings.

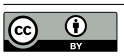
The constructivist approach of the unplugged activities gives students the opportunity to explore, and therefore the feeling that they invented the newly gained knowledge and skills themselves [1]. By focusing on teaching the underlying concepts of sorting with our unplugged activity and developing a genuine understanding of sorting algorithms, we give teachers the opportunity to build on these experiences in later computing, or specifically programming lessons.

We will mainly focus on comparison-based sorting with the possibility to re-enact well-known sorting algorithms like Bubble Sort, Selection Sort or Quicksort, but will take a look at related activities as well.

When asking students to perform any sorting algorithm unplugged the task is usually the same: 'Sort *something* by *some characteristic*', e.g., sort people by their height or age, sort numbers by their value or sort objects by their weight. Cards with letters can be used as explanatory aid [4]. Even colours can be sorted, e.g., by their hue based on the HSB-model [6]. As long as sortables have an internal order they can be used in various sorting activities, and if they are tangible, they can be used for unplugged activities. The question remaining is, whether there is a best 'something' or a best 'characteristic'.

As we want to address fostering Computational Thinking [13], we expand the above task: 'Sort something by some characteristic

*Work done while at TU Wien, Austria.



ACE 2026, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2352-0/26/02
https://doi.org/10.1145/3786228.3786257

as a computer would'. But what is the difference between sorting like a human and sorting like a computer? There are two significant differences: (1) Humans compare and arrange multiple items at once, even with multiple attributes such as colour, size or value, while a computer can only do pairwise comparisons at a time and needs repeated processes to address more attributes. (2) Swapping items looks like a seamless action executed by humans, while a computer needs a temporary placeholder, thus needing an extra step [11]. Considering these two differences the challenge using unplugged activities lies in encouraging students to act like a computer. Our material is designed in a way to let students experience the limited actions a computer has in connection with sorting algorithms.

2 Sorting Cards

Cards are a low-budget opportunity to create tangible sortables for unplugged sorting activities. The example we are presenting uses numbered cards, but most assumptions can be transferred to cards with other sortable attributes, such as letters, dates, or other items with a recognisable internal order.

Sorting based on comparison. Numbers are the most obvious sortable entity. K-12 students are able to sort numbered cards, depending on their range [9]. To encourage learners to sort them with pairwise comparison, one example is to flip the numbered cards upside down and add a rule: 'Reveal no more than two cards at a time'. This rule can be extended to a full set of rules to limit how cards are compared and moved, thus further encouraging students to act like a computer instead of using their natural holistic approach [5].

Sorting networks are a good way to make these comparisons visible. In the sorting network activity from CSunplugged every student gets one sortable and can only take one step, after a comparison took place [12]. The downside of these sorting networks is that they use unique algorithms that make use of parallelisation. This is not beneficial when planning to implement sorting algorithms without using parallelisation or when targeting well-known sorting algorithms.

Sorting vs. placing. Some positions of numbered cards are very predictable, e.g. the card with the number 1 on it. But we want the students to *sort* the cards instead of just *placing* them. Therefore, the final position of the sortables should be unpredictable by students. Using non-consecutive and larger numbers, e.g., cards with random numbers between 10 and 99, can help to make the final order less predictable [10].

Another approach can be seen in the sorting network activity by CS unplugged [12]. They offer various sets of cards with sortable images. They range from images out of a story to the development cycle of a butterfly or a tree[1]. When looking at one image, it is hard to tell where it belongs, but when comparing two images it is easy to put them in the correct order.

Embracing the unknown. Making the final position of the cards less predictable works quite well during a first run, but it leads us to the problem that students will remember the numbers after repeating the task. Thus, to further foster sorting behaviour like a computer, the characteristics need to be hard to remember. One option would be the use of stars as seen in Figure 1. Sorting the cards by the total number of star tips increases the difficulty to remember. Another option is using non-numerical properties, such as pictures of identical animals in different sizes [9]. They are easy to compare next to each other but it is not possible to remember a numerical size.

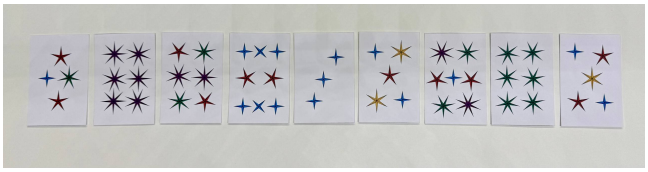


Figure 1: Sorting cards with stars indicating numbers.

There are other approaches to minimise this effect, e.g., swapping the cards after each run or drawing them randomly from a larger deck.

A group of university students at our institution developed an online tool based on an unplugged card sorting activity[2]. Each card shows a step in constructing a simple line drawing, with the sequence regenerated after every run, so the cards are always different. This prevents memorisation of card positions, but comparison remains easy, as a single differing line is enough to determine the order.

Hidden characteristics. Our experience using the cards with stars in class shows that the students still remember the order of the cards. This leads us to the conclusion that characteristics that are hard to remember are not enough, but instead the students must not be able to sort them globally or holistically at all. Thus, we need to use characteristics that are not visible. However, we still need a possibility to distinguish the sortables to verify the solution.

To hide the characteristic from the eye, invisible ink can be used [9]. This means that even when presented with the whole deck of cards, without a UV-light it is not possible to predict anything about the final order. This is especially useful when the comparisons are performed by different students. If only one student makes all comparisons, in the end they have seen all the different variations and thus it is only harder to overview the characteristic but not impossible.

Other approaches use digital devices to assign a value to an object. Dybboe et al. use Micro:bits to hold a value that is not accessible while sorting and is only revealed after the process of sorting is finished [3].

We follow the approach to replace the cards with tangible objects and sort them by an invisible characteristic.

3 Sorting Weights

When stepping away from visible characteristics, we still look for a characteristic that is constant and easy accessible such as the weight of a tangible object. Students can relate to an object having a specific weight and two objects with different weights can be easily compared using a scale.

[1] https://www.csunplugged.org/en/resources/sorting-network-cards/

[2] https://tools.lehr-lern-labor.info/SortLab

Figure 2: A set of sorting weights and matching scale

One possibility would be to use a weighing scale (e.g. a kitchen scale) and note the weight of each object. But this would contradict the idea of an invisible characteristic. Therefore, we use a balance scale. This fosters the idea of pairwise comparisons according to weight.

Using a balance scale to compare weights is not novel. Tim Bell et al. use a balance scale together with weights made of film canisters or milk cartons in their unplugged activity 'Lightest and Heaviest–Sorting Algorithms' [12]. Similarly, the exhibition 'Abenteuer Informatik' [4] includes an activity with one large balance scale and custom-filled paint cans that are used as weights.

Using the weight of objects as a characteristic for sorting and a balance scale to perform comparisons is a good solution for unplugged sorting activities. The objects have an internal order – from lightest to heaviest – and can be easily sorted by pairwise comparison.

Our remodelling of the activity from the original CS Unplugged activity [12] focusses on three aspects: (1) First, the original task uses eight weights and one balance scale per group. Buying those scales from vendors that offer school supplies or toy stores can be costly. (2) Second, the scales take up a lot of space to store and transport in school context. (3) Third, the original task uses film canisters as an example of do-it-yourself, which might be hard to get nowadays. We therefore present a version that is (1) low-cost, (2) space-saving and (3) easy to craft in today's school context.

4 Finding the Right Balance

As an alternative to store-bought balance scales and self-crafted weights, we introduce 3D-printable sorting weights and a matching balance scale as seen in Figure 2. The cubes are printed with different infill settings, resulting in pairwise different weights. The balance scale is as simple as a seesaw and works with the torque resulting from the weight difference between the compared cubes. Due to the fixed shape of the weights a more complex design like a pan balance or a roberval balance is not necessary.

4.1 The use case

Our aim is to provide an easy to use and easy to produce classroom-sized set of balance scales and matching sorting weights. Therefore, the components need to be small enough to fit on students' desks and in teachers' bags and durable enough to outlive multiple generations of students. This influenced the technical design described in sections 4.2 and 4.3. Furthermore, they have to be compatible with existing teaching approaches. Therefore, we focus on the ability to sort the weights using a well-known sorting algorithm. We focus on *Selection Sort*, *Bubble Sort* and *Quick Sort*, but there are other sorting algorithms that can be used to sort the weights as well. To take the differences between plugged and unplugged sorting into account, we use slightly adjusted versions of these well-known algorithms.

Selection Sort is very intuitive for students. If they have no idea on how to start, they can be asked to find the heaviest cube and to put it aside. By repeating this process and lining up the respective heaviest cube they eventually end up with a sorted array of weights. Bubble Sort can be easily implemented in-place. While Selection Sort always needs the same amount of comparisons, Bubble Sort shows a noticeable difference in the number of comparisons needed between best and worst case.

In our experience five weights are enough to grasp the concept of Selection Sort and Bubble Sort. However, using more weights makes the task of sorting more challenging and also more interesting, as it allows more initial permutations and therefore more variation in the actual sorting process.

Quick Sort relies on the principle of divide and conquer. We usually do not try to implement it in place, as it is hard to keep track of the different pointers. Instead we move the weights and rotate them when they are placed in their final position. In our experience, at least eight weights should be used to perform this algorithm. This ensures that in the best case one of the groups from the first split has to be split at least two more times.

Furthermore, we chose to use an odd number of weights. Some students initially experimented by comparing all weights in pairs and tried to build a solution from that, which is not effective. With an odd number of weights, the leftover weight disrupts this strategy, prompting reflection and guiding students toward a more promising approach.

4.2 The Weights

We designed the weights to be cube-shaped. This allows for easy positioning and prevents them from rolling off the table. Every cube has a unique icon embossed on the top side, the other sides are blank. This allows us to flip the cubes upside down and thus make them indistinguishable from all visible sides. We used icons instead of numbers or letters, as they have no internal order. Instead, they are assigned randomly and can be used to verify the result or to communicate an initial pre-sorting.

The first version of the weights did have a side length of 5 cm. This turned out to be a good size for demonstration in front of the class but is too big to be used by individual students. Therefore we halved the size and ended up with a side length of 2.5 cm. The cubes are still big enough to grip them easily, yet small enough to be used on small desks and to store and transport them conveniently.

We wanted the set to be fully 3D-printable without the need for assembling or embedding additional objects (such as small metal parts) in the print to increase weight. With the limited size, the heaviest cube could weigh about 20 g max using 100% infill. For the lightest cube we had to make sure that the shell still printed well. We ended up with nine cubes between 5 g and 17 g using between 5% and 90% infill, depending on the used 3D-printer and the specific settings. In theory, two more cubes could be added. The weight-difference of only 1.5 g is hardly perceptible by hand but detectable by the matching balance scale. A narrower spacing proved to be problematic as the balance scale was not sensitive and reliable enough, any more.

4.3 The Scale

For the scale we used the advantage that the weights all have the same shape. This means that we can use a simple balance scale using the concept of a seesaw as we can assure the same distance of the center of mass from each weight to the fixed point of the scale. Therefore, the resulting torque that tilts the scale depends only on the weight difference between the two compared weights.

Alternatively, the 3D-printed weights can be used on a store-bought comparison scale. But roberval balances tend to oscillate a lot, which is a problem with small weight differences. This makes it tedious to get a result of the comparison. In contrast, the simple 3D-printed balance scale just tilts to one side and doesn't oscillate at all. This leads to a faster and more accurate result while allowing for smaller weight differences of the weights. Despite the small weight difference, our experiments show that the matching scale works very reliably. When using weights printed with different printers or different filament, there are some edge-cases where the resulting torque is not enough to tilt the scale to the other side. But when manually moved to the middle position, it still works reliably. This shows that we have reached a natural limit of the accuracy of a 3D-printed simple balance scale in this size.

An entire set of one scale and nine cubes weighs about 115g, which means that with one 1kg-spool of filament we can produce eight sets of scales and weights. Therefore, one spool of filament is enough to provide a set to every group of four in a class with 32 students in total.

5 Experience

Until now, the presented sorting weights have been used in different outreach programs in Germany, Austria and India. These outreach programs are characterised as one-time interventions, meaning that the students do not take part in activities on a regular basis, in particular not in a school-like setting. However the programs differ in their length and their intention. Despite the lack of longitudinal data, the repeated and consistent engagement across diverse contexts provides practical evidence for the accessibility and motivational potential of the activity.

5.1 Hands-On-Exhibit

The sorting weights have been used as a hands-on-exhibit as part of different interactive exhibitions. These interactive exhibitions do not focus on computer science but try to engage students in science in general. Our booth focuses on computer science and presents multiple unplugged activities, including the sorting weights. Hereby, our aim is to foster students' motivation toward computer science, to introduce students to meaningful computer science topics and to highlight the advantages of unplugged computer science activities. In this setting, students pass by the booth with the exhibit and decide on their own whether they want to interact with the exhibit or not. If they decide to interact with the exhibit, student assistants guide them through the task while taking the students' prior knowledge and personal interests into account. Since introducing the sorting weights as a hands-on-exhibit, more than one hundred students of all age groups have participated in the activity.

Usually the initial task is to sort the weights from lightest to heaviest using the scale. If more than one student approaches the booth at a time, the activity is turned into a competition and the students have to sort the weights as fast as possible. Primary school students sometimes need help with the task, mostly because they struggle with keeping the sorted weights organized and tend to mix them up during the process of sorting. Secondary school students usually succeed in sorting the weights, although some minor mix-ups might occur. Some older students still struggle with keeping the weights organized, but they are usually able to resolve mix-ups independently – by restarting if necessary.

As a hands-on-exhibit, the sorting weights draw attention and motivate students of all ages to experiment with the weights and the scale. Even though some students enjoy stacking up the weights to build a tower on top of the scale, most students grasp the concept of sorting the weights using the scale quite intuitively and are able to complete the task using their holistic approach.

5.2 Algorithm Workshop

In workshops focusing on algorithms, the sorting weights were used as an intervention to introduce different sorting algorithms. Before and afterwards, the students worked with cards, developing their own sorting strategies. Using the sorting weights, they were introduced to Bubble Sort and Selection Sort, but still had to transfer their understanding of the newly introduced algorithms to the task of sorting cards. To make the presented sorting algorithms easier to follow, we used weights in different colours. As the time did not allow for many repetitions, it was not a big problem that the students remembered the colours. The transition from using numbered cards to using weights and vice versa, supports students in transforming their intuitive, holistic strategies into more formal, computer-like approaches.

In multiple workshops over one hundred students aged 11 to 12 took part in the presented activity (see [8]).

5.3 Enrichment Activity

The sorting weights were also used as enrichment activities within multi-day computer science courses. This means the students attend a multi-day course on a specific topic in the field of computer science. In addition to the activities about that topic, they are introduced to other fields of computer science by engaging in enrichment activities. E.g. the sorting activity was conducted as an enrichment activity within a one-week program where 13 girls between 13 and 17 years mainly focused on building their first website. Until today

about 30 students from age 13 upwards took part in enrichment activities involving the sorting weights and scale.

In this activity every student got their own set of sorting weights and we challenged the students to sort the weights as fast as possible. Next, they had to count the number of comparisons and we discussed the difference between the various approaches and identified well-known sorting algorithms the students had intuitively applied. This started a discussion about the advantages and disadvantages of different algorithms and the differences between sorting by humans and computers. Therefore, the activity not only supported conceptual understanding but also encouraged reflection on algorithmic efficiency and strategy, contributing to students' computational thinking.

6 General Observations

Most students grasp the concept of sorting weights almost instantly. A few students still struggled when using the sorting weights the first time. The most common problem were mix-ups of weights during the sorting process.

While most students engaging in the sorting activities have a prior understanding of the concept of weighing, one student did not understand the transitivity of weighing, meaning they figured out that weight A is heavier than weight B and weight B is heavier than weight C, but did not understand that this implies weight A being also heavier than weight C. Therefore, more complex sorting algorithms like Quick Sort were just not possible to understand at that time.

Furthermore, some students used their perception to presort the weights to 'save' some comparisons. At this point, we introduced additional rules like 'icons need to face down while sorting' or 'lifting two weights counts as comparison.'

7 Technical Limitations and Future Work

As the outreach-projects are one-time activities we have no data on long-term impact, yet. However, given the well-established role of sorting in computer science education and the alignment with existing unplugged approaches, it is reasonable to assume comparable long-term benefits. Future work might explore this further.

While the current design offers a reliable hands-on experience,the weight difference of the 3D-printed weights is still noticeable but reaches the limit of the 3d-printable weights and scales. To improve precision and enable automated tracking of comparisons, future iterations might incorporate NFC tags or similar technologies. This way, instead of using the weight of the cubes we can use a 'simulated weight' that is saved in the tag and therefore can not be perceived manually. This approach would require a more advanced scale capable of reading and comparing digital tags, but might mitigate some shortcomings of the presented weights and scales while still keeping the hands-on character of the unplugged activity.

8 Conclusion

Sorting Algorithms are an important part of early computer science education and providing an unplugged activity offers great potential for understanding the concept of sorting and the differences of various sorting algorithms. The presented sorting weights with the corresponding scale offer easy access to tangible educational material. Educators only need access to a 3D-printer and can produce their own compact classroom-sized set of sorting weights and balance scales. Furthermore, these weights can be used in different settings and can be easily integrated in existing or newly developed classroom activities while combining advantages of previously existing teaching materials.

The files required for printing are freely available under an open-source license at research.lehr-lern-labor.info/sorting-weights.

References

[1] Tim Bell, Paul Curzon, Quintin Cutts, Valentina Dagiene, and Bruria Haberman. 2011. Overcoming Obstacles to CS Education by Using Non-programming Outreach Programmes. In *Informatics in Schools. Contributing to 21st Century Education*, Ivan Kalaš and Roland T. Mittermeir (Eds.). Lecture notes in computer science, Vol. 7013. Springer Berlin Heidelberg, Berlin, Heidelberg, 71–81. doi:10.1007/978-3-642-24722-4{_}7

[2] Tim Bell and Jan Vahrenhold. 2018. CS Unplugged—How Is It Used, and Does It Work? In *Adventures between lower bounds and higher altitudes*, Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger (Eds.). Lecture notes in computer science, Vol. 11011. Springer, Cham, 497–521. doi:10.1007/978-3-319-98355-4{_}29

[3] Maja Dybboe, Magnus Høholt Kaspersen, Johanne Birkkjær Bjerrum, and Marianne Graves Petersen. 2025. Bit:sort : Bringing Tangible Computing to Computer Science Unplugged to Support Children's Algorithmic Explorations. In *Proceedings of the 24th Interaction Design and Children (IDC '25)*. Association for Computing Machinery, New York, NY, USA, 429–443. doi:10.1145/3713043.3728854

[4] Jens Gallenbacher. 2012. *Abenteuer Informatik: IT zum Anfassen - von Routenplaner bis Online-Banking* (3. aufl ed.). Springer Spektrum, Berlin Heidelberg.

[5] Inf-Schule.de. 2024. Entwicklung von Sortieralgorithmen. https://inf-schule.de/algorithmen/standardalgorithmen/sortieren/entwicklung.

[6] Tobias Kohn and Dennis Komm. 2018. Teaching Programming and Algorithmic Complexity with Tangible Machines. In *Informatics in Schools. Fundamentals of Computer Science and Software Engineering*, Sergei N. Pozdniakov and Valentina Dagienė (Eds.). Springer International Publishing, Cham, 68–83. doi:10.1007/978-3-030-02750-6_6

[7] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023*. ACM, New York, NY, USA. doi:10.1145/3664191

[8] Martina Landman. 2026. *Algorithmic Problem Solving in Unplugged Computer Science Outreach Activities*. Ph. D. Dissertation. TU Wien.

[9] Martina Landman, Sophie Rain, Laura Kovács, and Gerald Futschek. 2023. Reshaping Unplugged Computer Science Workshops for Primary School Education. In *Informatics in Schools. Beyond Bits and Bytes: Nurturing Informatics Intelligence in Education*, Jean-Philippe Pellet and Gabriel Parriaux (Eds.). Springer Nature Switzerland, Cham, 139–151. doi:10.1007/978-3-031-44900-0_11

[10] Martina Landman. 2025. Sortieren – wie ein „Computer", aber ohne Computer. https://www.netidee.at/algorithmic-problem-solving-unplugged-computer-science-outreach-activities/sortieren-wie-ein

[11] Piyanuch Silapachote, Ananta Srisuphab, Apirak Hoonlor, and Thanwadee Sunetnanta. 2024. Mastering basic Sorting Algorithms through Computational Thinking Activities for Everyone. In *2024 IEEE Global Engineering Education Conference (EDUCON)*. 1–5. doi:10.1109/EDUCON60312.2024.10578701

[12] Tim Bell, Ian H. Witten, and Mike Fellows. 2015. *CS Unplugged: An enrichment and extension programme for primary-aged students*. https://classic.csunplugged.org/documents/books/english/CSUnplugged_OS_2015_v3.1.pdf

[13] Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. doi:10.1145/1118178.1118215