

The Great Number Hunt (Sorted)

- Duration: 45 minutes
- Ages 8 to 10: Lesson 4

Printables



Number Hunt
Includes unsorted and sorted variants.

Classroom resources

- Pencils

⊕ Learning outcomes



Students will be able to:

- Describe how the time taken grows with the size of the input, and most importantly how it grows in different ways for two different algorithms.
Computational Thinking: Generalising and Patterns
- Describe how to compare number values for equality and inequality (greater than, less than).
Mathematics: Numeracy
- Explain how they used decomposition to divide and conquer when doing a binary search.
Computational Thinking: Decomposition
- Explain the range of the number of guesses for unsorted lists compared to sorted lists.
Mathematics: Statistics
- Identify search algorithms for sorted and unsorted lists (sequential and binary search).
Computer Science: Algorithms

Key questions



How would you look for a book in a library if the books were sorted in alphabetical order? Is that easier than if they were out of order?

Potential answers

Guide students to discuss the algorithm or process that they would use to find a value in a sorted list and how that would be different in an unsorted list. Help them to recognise that the best way to search the sorted list is to do a binary search: it does matter what order they actually look at the books.

Lesson starter

As with lesson 3, on each sheet are 31 number boxes with lots of money in them. All money has a serial number on it and in each of the number boxes all the notes start with the same serial number. The only problem is there is only one number box

that has the real money in it, the rest are fake. You know what the serial number on the real money is, and your challenge is to find the number box with the real money in it, in the least amount of guesses.

Guide students through how the game will work by following the instructions on the [Number Hunt](#) (although this time the numbers will be in sorted order).

⊕ Mathematical Links

This is a teachable moment for the class to discuss different strategies to ensure the smallest number of guesses is used. It's not essential that they use the "best" method (starting with the middle item), since they will learn the hard way that this would have been a good idea, but it may save some frustration if they become aware of the strategy in advance.

Statistics: Gather the raw data of the number of guesses made and graph it on a bar chart. Repeat the game a number of times to see what the pattern of data is.

Number Knowledge: Read and say numbers between 100 and 9999.

Lesson activities

1. Organise your class into pairs.
2. They need a hard surface to write on and a pencil or pen (or whiteboard pen if you have laminated the sheets).
3. Have students seated back to back.
4. Have everyone come up and get a [Number Hunt printable](#).
5. Go through the instructions and set up their games.
6. Play the game and have students add up how many guesses it took each of them to find the number box with the correct number in it.
7. Add this data to either a class sheet to graph on paper or add it to a spreadsheet that also displays the graph and what the graph grow as more data becomes available.

Applying what we have just learnt



The method you have been using is called binary search - if the values are in order then we can eliminate half of the numbers with each guess until we find the one we want. This is an incredibly efficient algorithm that we can use for searching a sorted list.

It also shows how sometimes algorithms aren't strongly affected by how big the problem gets - we can search billions of items (e.g. every web site we can find) using this method; if we had 1,000,000,000,000 number boxes, you'd only need to open 40 of them to find a particular one; and doubling the number of boxes only adds one more question to finding a value. This is a pleasant contrast to some other problems where we don't have such efficient algorithms.

Lesson reflection

What is the algorithm for a binary search for the Number Hunt game?

Seeing the Computational Thinking connections

Throughout the lessons there are links to computational thinking. Below we've noted some general links that apply to this content.

Teaching computational thinking through CSUnplugged activities supports students to learn how to describe a problem, identify what are the important details they need to solve this problem, break it down into small logical steps so that they can then create a process which solves the problem, and then evaluate this process. These skills are transferable to any other curriculum area, but are particularly relevant to developing digital systems and solving problems using the capabilities of computers.

These Computational Thinking concepts are all connected to each other and support each other, but it's important to note that not all aspects of Computational Thinking happen in every unit or lesson. We've highlighted the important connections for you to observe your students in action. For more background information on what our definition of Computational Thinking is [see our notes about computational thinking](#).

⊕ Algorithmic thinking



Binary search works by applying the Divide and Conquer process; we repeatedly check the centre number box and deduce which boxes can then be eliminated, and which ones could still potentially contain the real money you are hunting for. This can be given as an algorithm.

By describing this method with the same algorithm as in lesson 2 a computer or person can follow it without having to think about how to actually do the task.

Examples of what you could look for:

Who are the students who not only can explain the exact process to find the real money, but are also the students who don't deviate from that process?

⊕ Abstraction



As in lesson 2 and lesson 3, we can ignore many details about the items we are searching for.

Also, as discussed in lesson 2, we can use the divide and conquer approach for more problems than just searching through an ordered list of numbers. We can use it to search through any set of objects that can be placed in ascending order (such as words in alphabetical order).

Examples of what you could look for:

What questions are students asking about the number boxes? Organise them into useful questions and no as useful questions. Which students can identify quickly what the relevant information is?

⊕ Decomposition



Binary search is a classic method of divide and conquer and is entirely about decomposition. Refer to lesson 2.

Examples of what you could look for:

Who are the students who are able to break the problem down into steps and then explain why each step is important?

⊕ Generalising and patterns



The key pattern to recognise in this activity is the process of eliminating half the possible cards by only looking in one, and that this is repeated over and over to accomplish the task.

Like we talked about in the lesson plan, the divide and conquer strategy is a pattern that appears frequently in computer science, and also in real life! It is an efficient and logical way of attacking many different problems where you are searching for something in a group of objects that have different identifying features.

Examples of what you could look for:

Who are the students who quickly identified the pattern?

⊕ Evaluation



Students will be evaluating binary search in the same way they evaluated the divide and conquer method in lesson 2.

Students can evaluate how well the divide and conquer method works by looking at how many marbles (or whatever payment you decide to use) they have left at the end of the activity. Older students can further examine the efficiency of this algorithm by calculating the maximum number of checks it would make for a different numbers of cards. You could compare this to the number of checks that a sequential search would need, and how these numbers change as you increase the number of cards.

Examples of what you could look for:

Who are the students who can explain the strengths and potential problems of using a binary search to find data?

⊕ Logic



To search through the number boxes and guarantee they will find the real money in a low number of guesses students must use the binary search algorithm rather than using a sequential search or just choosing randomly based on where they think the money might be. This is because they need to eliminate as many boxes as possible with each guess. Students might think if there are 100 boxes and the number they are looking for is 90 then the logical choice would be to look in a boxes near the right end of the list first, instead of the middle, because they think 90 is likely to be near the 90th box. But this isn't actually a more logical choice than dividing the list in half with binary search if you don't know the range of serial numbers on the money in the boxes, and binary search is so fast that trying to be clever about where a value is doesn't help much even if you're right. Which students can explain why the logical decision is to always stick with binary search? A good question to prompt them with is:

- You don't know what the range of serial numbers on the money is, it could be anything! If they were somewhere between 1 and 99,999,999 and you have 100 number boxes would it be a good choice to guess where the money might roughly be? Or should you stick to binary search?

Examples of what you could look for:

01	04	05	08	12
16	19	22	23	27
30	34	38	39	42
47	50	60	61	66
68	74	77	80	96

Which students instinctively go for the middle square when searching? They are likely logical thinkers who can deduce that since the numbers are sorted then the middle square will tell them the most useful information.