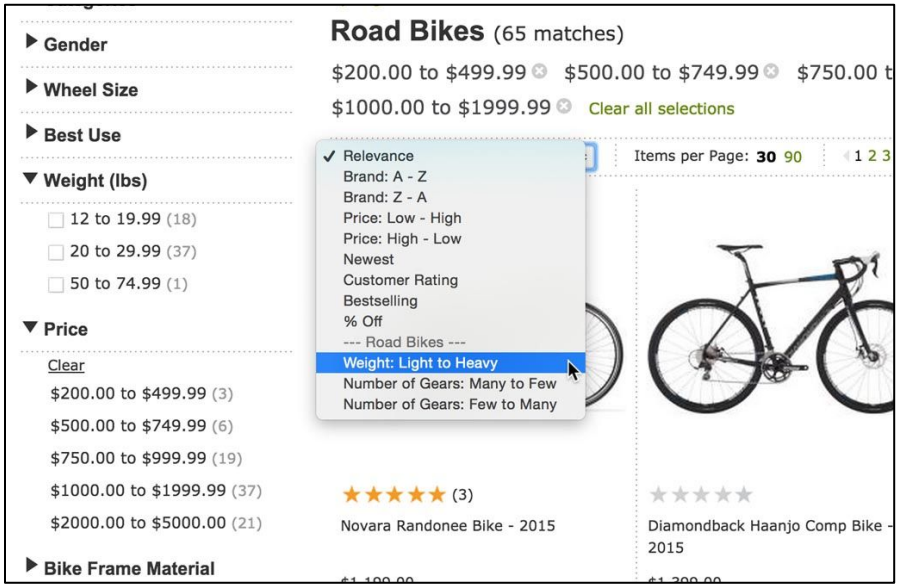
Unit 8 - Lesson 4
Selection Sort

Computer Science A

💡 Discuss:

How do you use the **"Sort By"** feature on an **e-commerce site** like Amazon?

What are some of the **different options** offered to **sort** the products?



🎯 Lesson Objectives

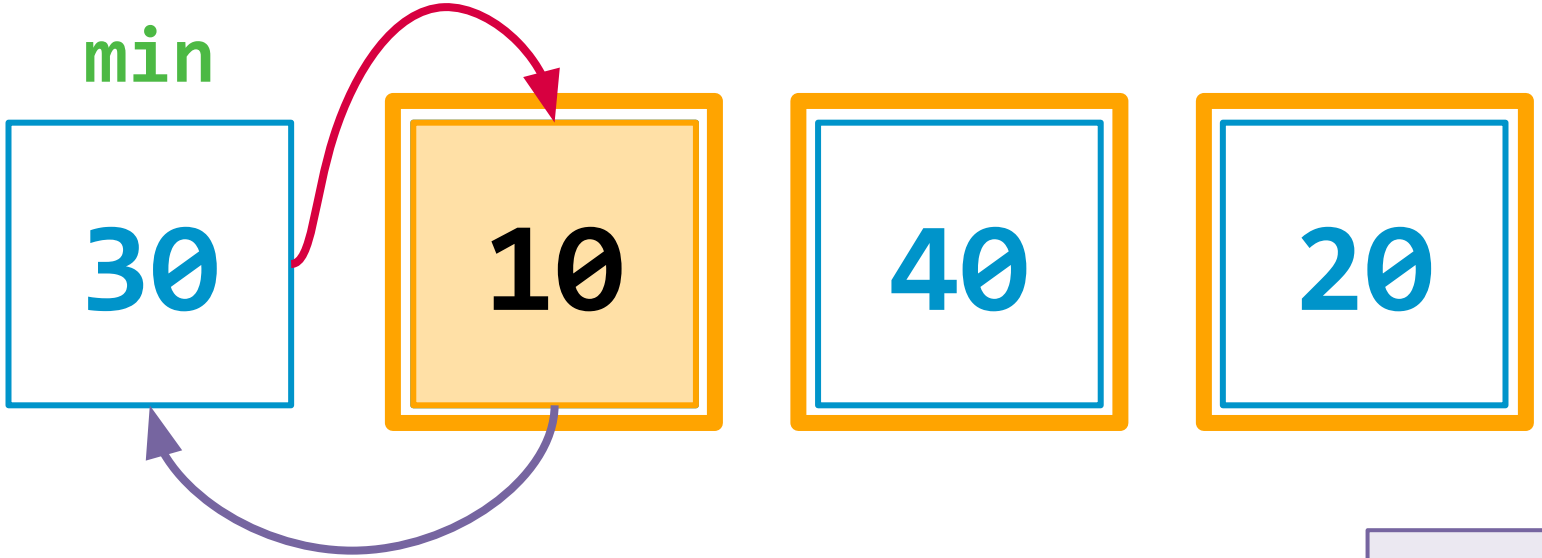
By the end of this lesson, you will be able to . . .

- Analyze the efficiency of the selection sort algorithm
- Explain the functionality of the selection sort algorithm

🚀 Question of the Day

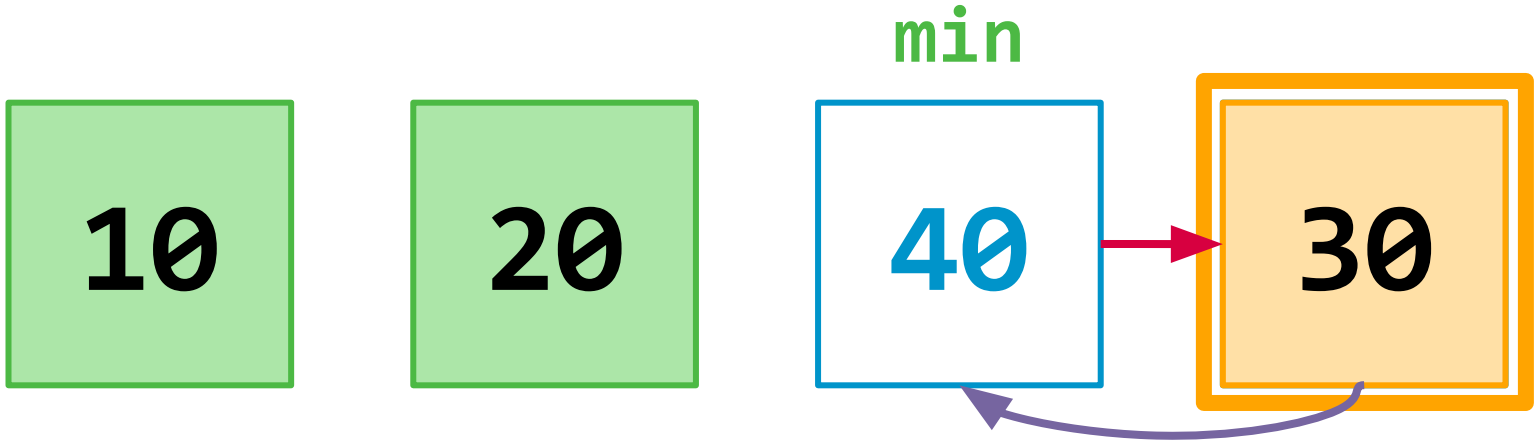
How can I sort elements in a data structure?

The `selection sort` algorithm selects the **smallest element** from an unsorted array in **each iteration** and places that element at the **beginning** of the unsorted array.

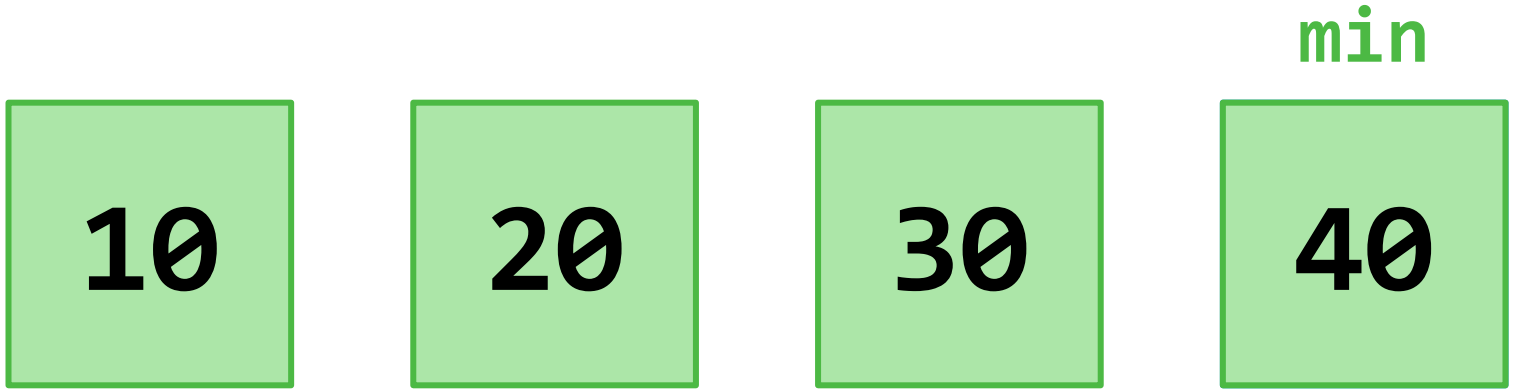


The `selection sort` algorithm selects the **smallest element** from an unsorted array in **each iteration** and places that element at the **beginning** of the unsorted array.

The `selection sort` algorithm selects the **smallest element** from an unsorted array in **each iteration** and places that element at the **beginning** of the unsorted array.



The **selection sort** algorithm selects the **smallest element** from an unsorted array in **each iteration** and places that element at the **beginning** of the unsorted array.



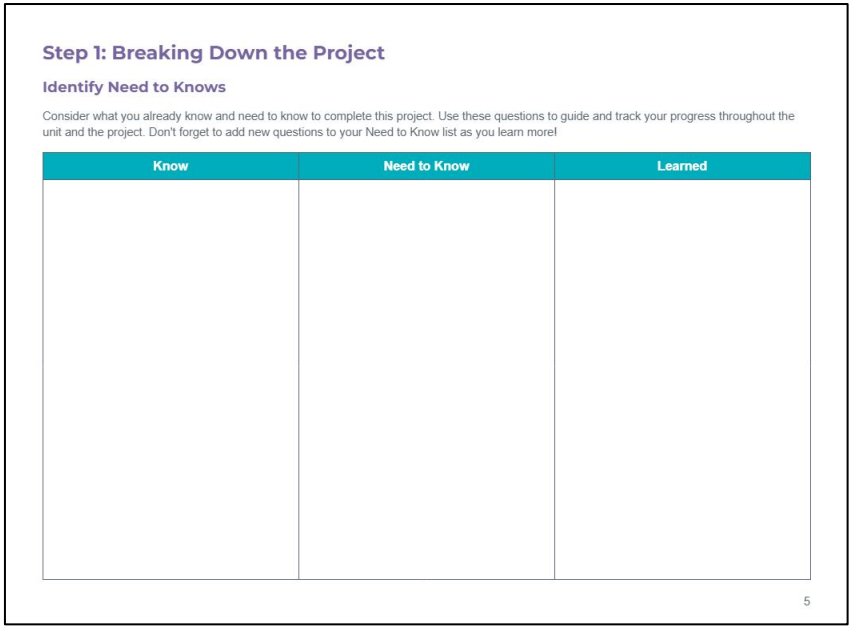
Selection Sort Algorithm

```
set min to location 0
search for the smallest value in the list
  swap with value at location min
  increment min to point to next element
  repeat until list is sorted
```

✅ Do This:

Revisit your **Need to Knows**!

- Check off **answered questions** in the **Need to Know** column.
- Add what you have **learned** and **answers to any questions** in the **Learned** column
- Add any **new questions** to the **Need to Know** column



Step 1: Breaking Down the Project

Identify Need to Knows

Consider what you already know and need to know to complete this project. Use these questions to guide and track your progress throughout the unit and the project. Don't forget to add new questions to your Need to Know list as you learn more!

Know	Need to Know	Learned

5

💼 Key Vocabulary

- **selection sort:** a sorting algorithm that selects the smallest element from an unsorted array in each iteration and places that element at the beginning of the unsorted array

Unit 8 - Lesson 5
Insertion Sort

Computer Science A

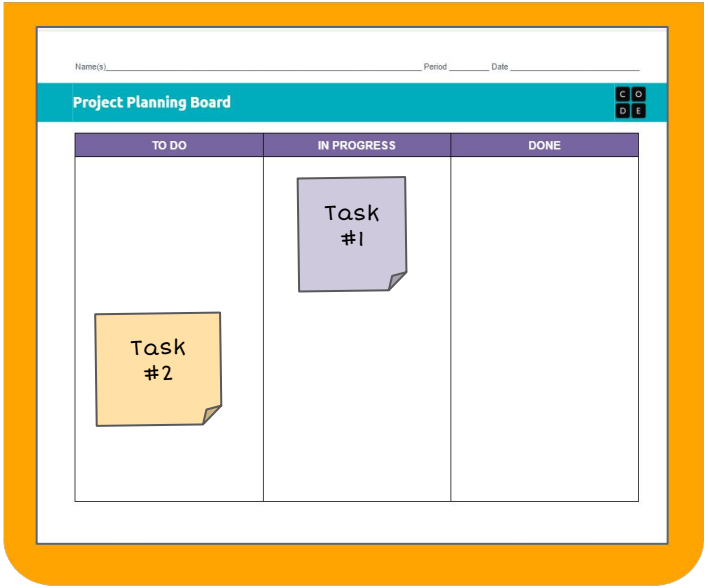
🎯 Benchmark #1: Due Lesson 6

- Brainstorm project ideas and goals
- Decompose the problem to identify the classes and methods you will need to implement
- Obtain and implement feedback from peers

✅ Do This:

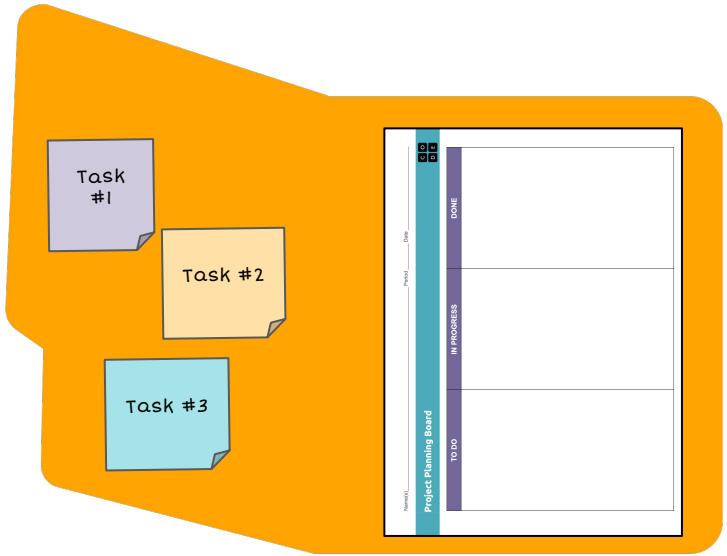
Move the task you will work on to the **IN PROGRESS** column of your Project Planning Board.

Work on your Creative Coding with the Console Project.



✅ Do This:

Update your **Project Planning Board** and **Project Backlog** with any tasks you completed, changed, or added.



🎯 Lesson Objectives

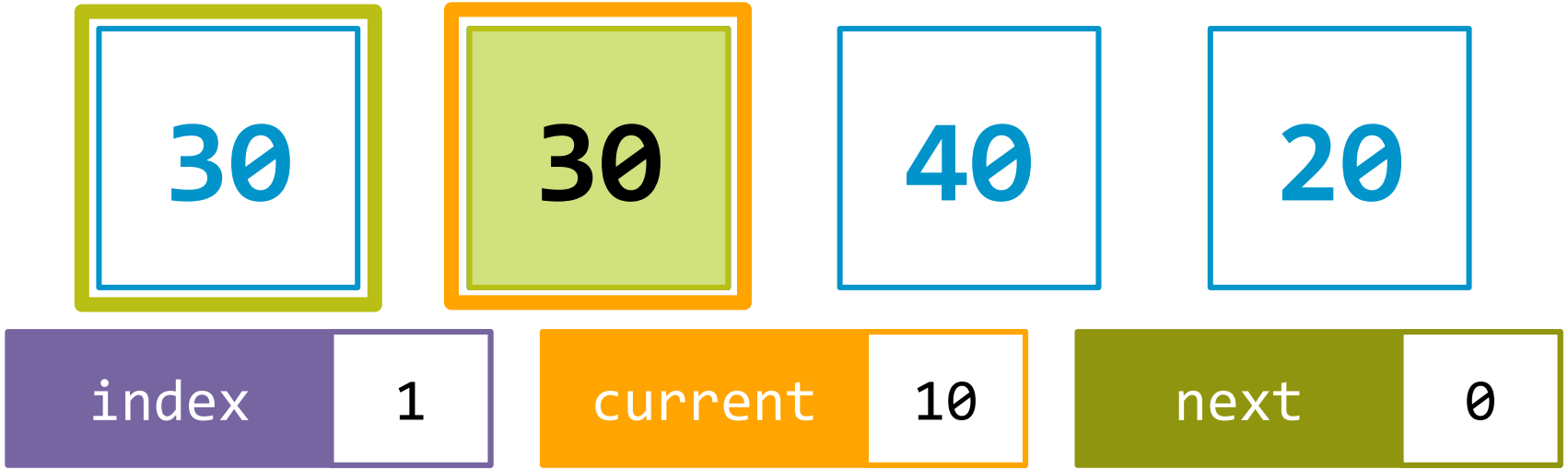
By the end of this lesson, you will be able to . . .

- Compare the efficiency of the insertion sort algorithm with the selection sort algorithm using execution counts
- Explain the functionality of the insertion sort algorithm

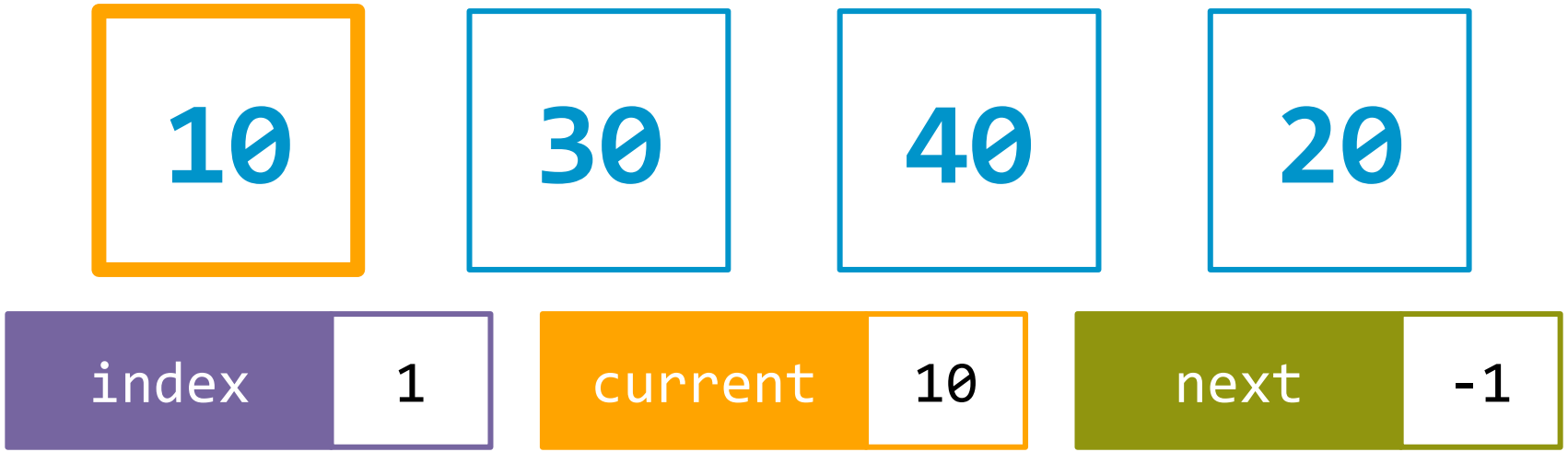
🚀 Question of the Day

How does the insertion sort algorithm compare in efficiency with the selection sort algorithm?

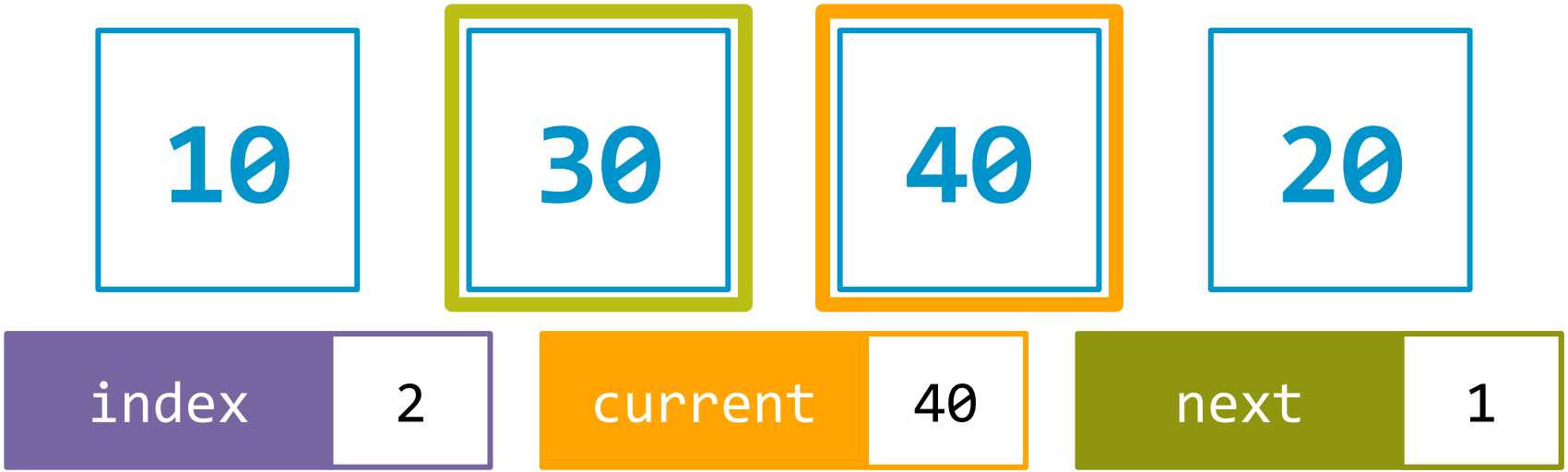
The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



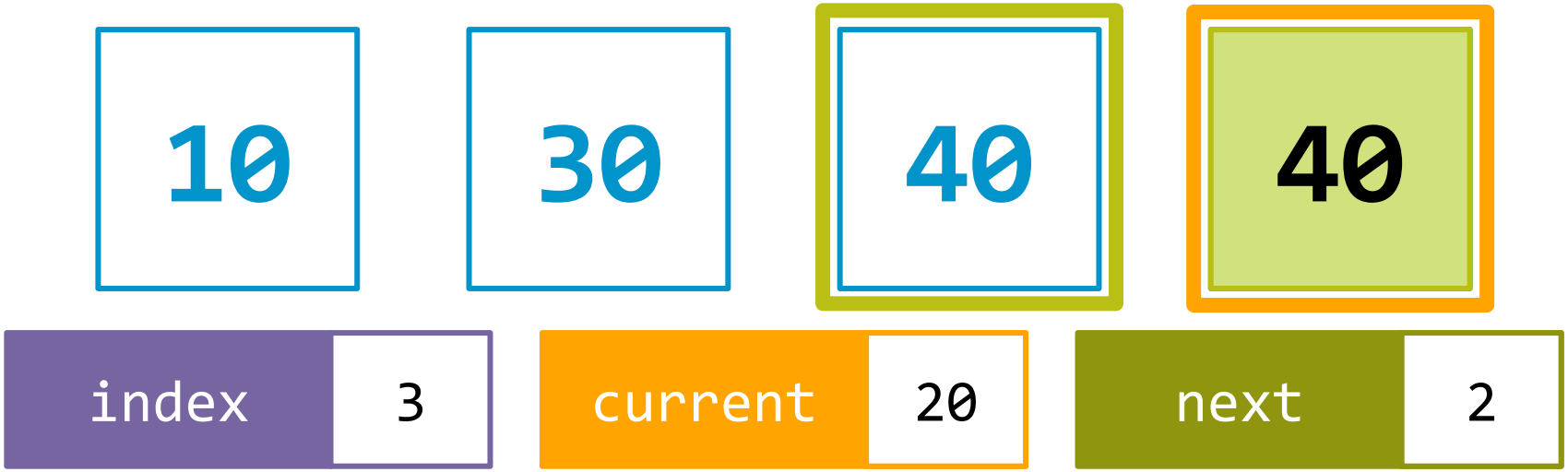
The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



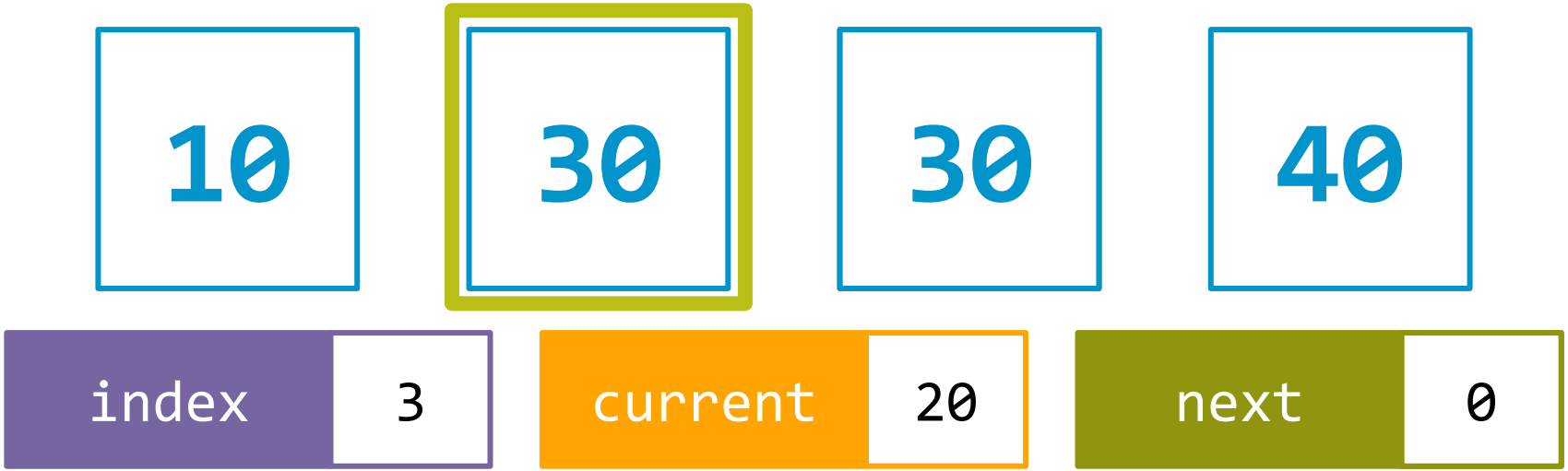
The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



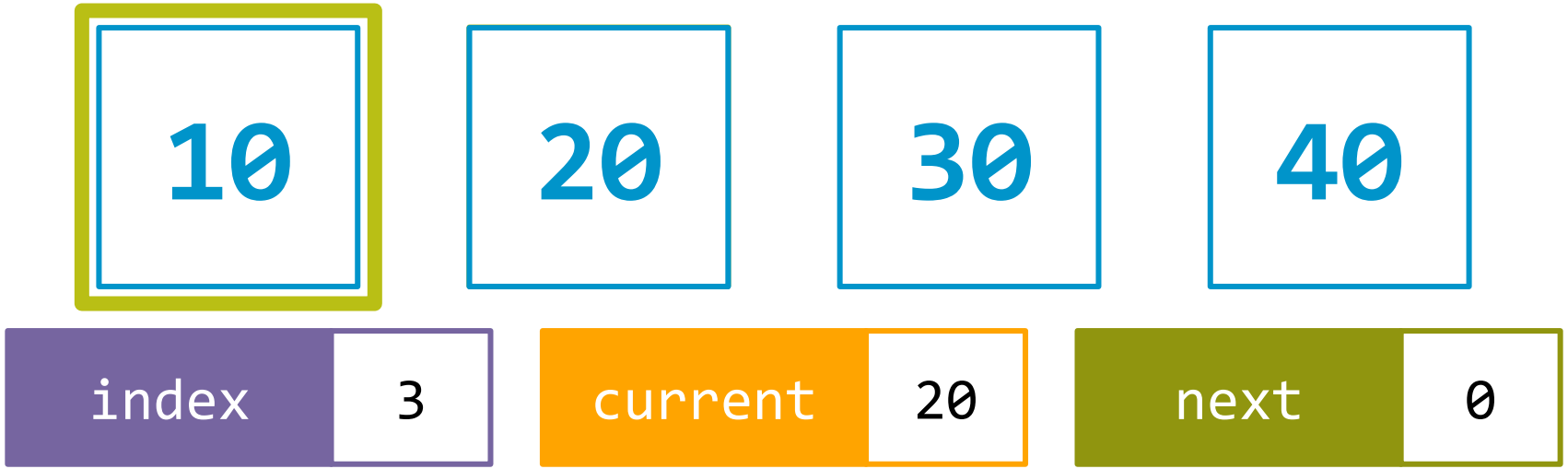
The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



The **insertion sort** algorithm **shifts each item** in a list one at a time to the **correct position** in the sorted portion of the list.



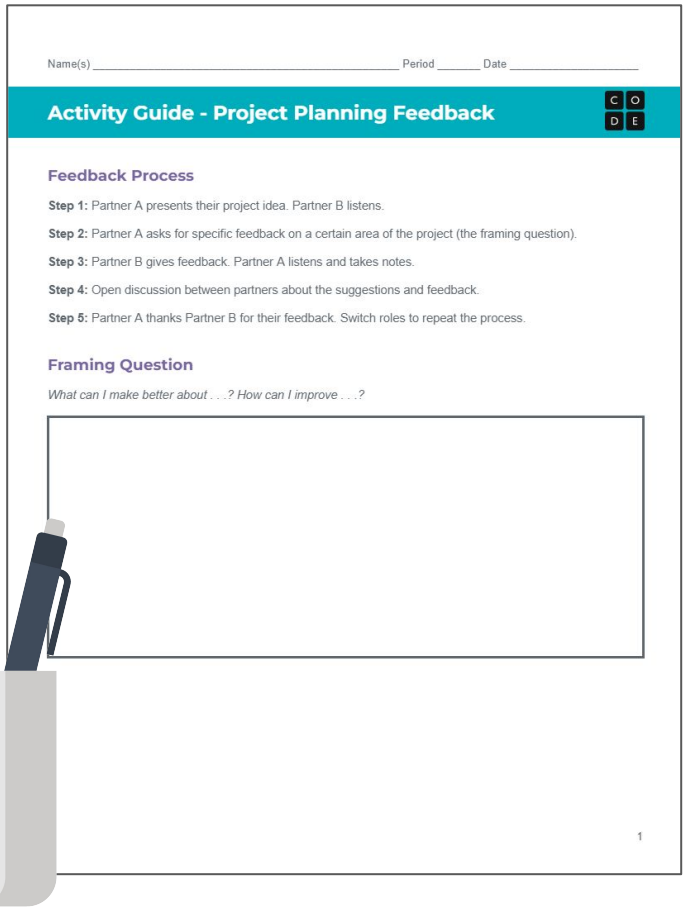
Insertion Sort Algorithm

```
loop list from index 1 to the end of the list
  set current to the value at index
  set next to index - 1
  while next is greater than or equal to 0 and the value at next is greater than current
    set the element at next + 1 to the value at next
    decrement next
  set the element at next + 1 to current
```

Project Planning Feedback

📍 **You and your partner should have:**

- Project Planning Feedback handout
- pen / pencil



Name(s) _____ Period _____ Date _____

Activity Guide - Project Planning Feedback

Feedback Process

Step 1: Partner A presents their project idea. Partner B listens.

Step 2: Partner A asks for specific feedback on a certain area of the project (the framing question).

Step 3: Partner B gives feedback. Partner A listens and takes notes.

Step 4: Open discussion between partners about the suggestions and feedback.

Step 5: Partner A thanks Partner B for their feedback. Switch roles to repeat the process.

Framing Question

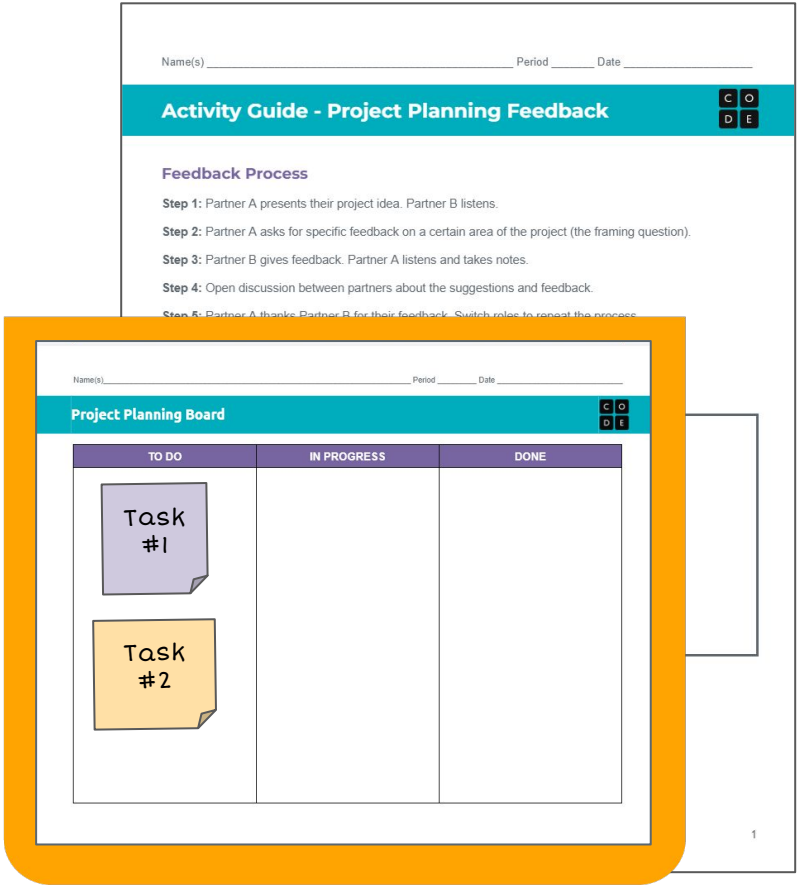
What can I make better about . . .? How can I improve . . .?

1

✅ Do This:

Respond to the prompt on your Project Planning Feedback handout.

Post the tasks for the second benchmark in the **TO DO** column of your Project Planning Board.



💼 Key Vocabulary

- **insertion sort:** a sorting algorithm that shifts each item in a list one at a time to the correct position in the sorted portion of the list

Unit 8 - Lesson 6
Merge Sort

Computer Science A

✅ Do This:

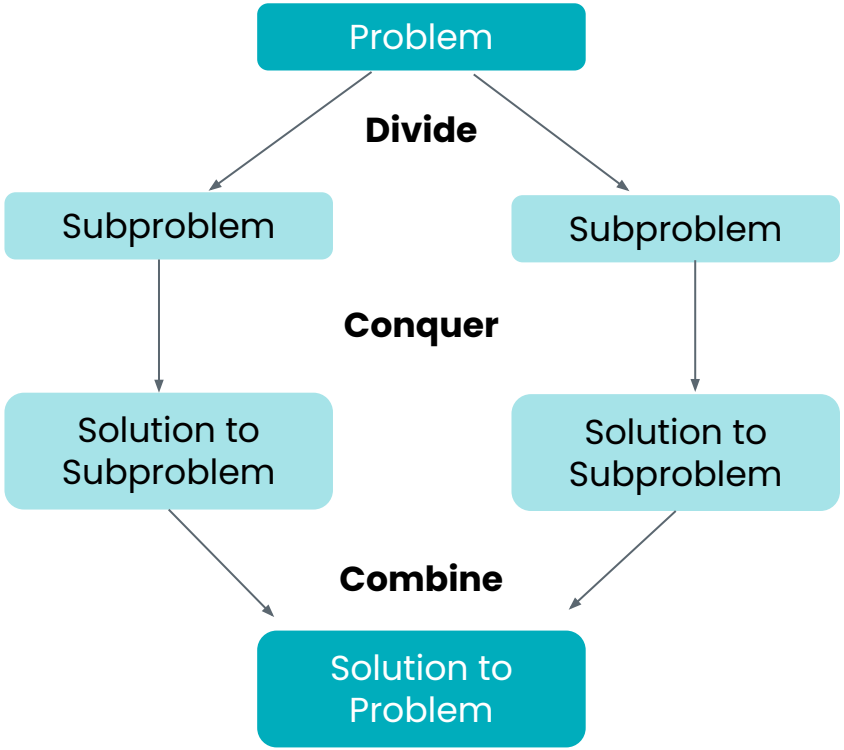
Write an algorithm for **counting loose change**.

3 * 25¢ = 75¢

7 * 5¢ = 35¢

2 * 1¢ =2¢

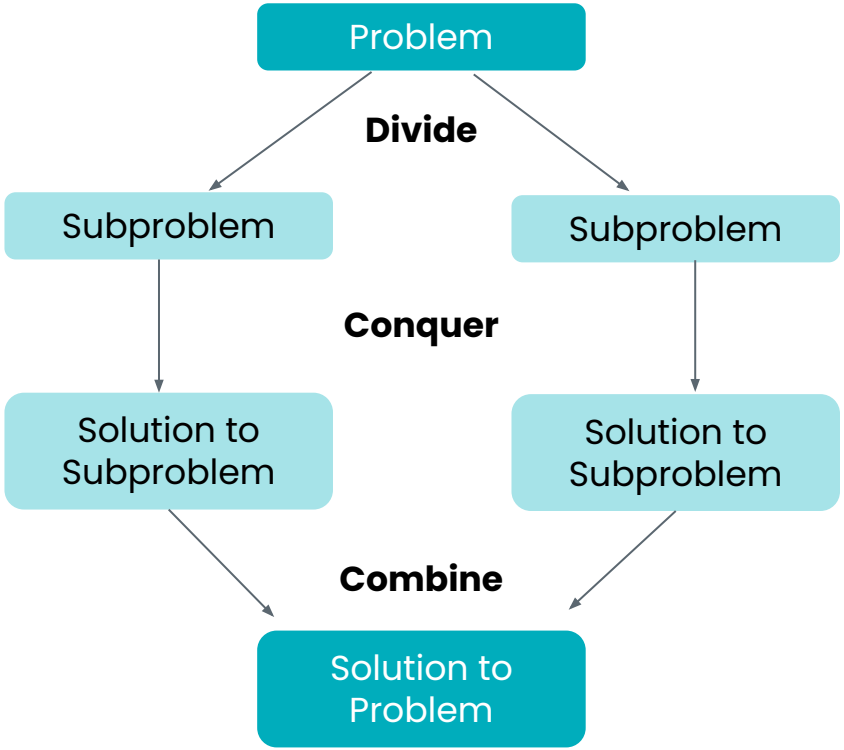
75¢ + 35¢ + 2¢ =
$1.12



Divide-and-conquer is the process of **dividing a problem** into smaller problems, **solving the smaller problems** independently, and then **combining the solutions** to solve the original problem.

💡 Discuss:

How do you think **sorting algorithms** can use a **divide-and-conquer** approach?



🎯 Lesson Objectives

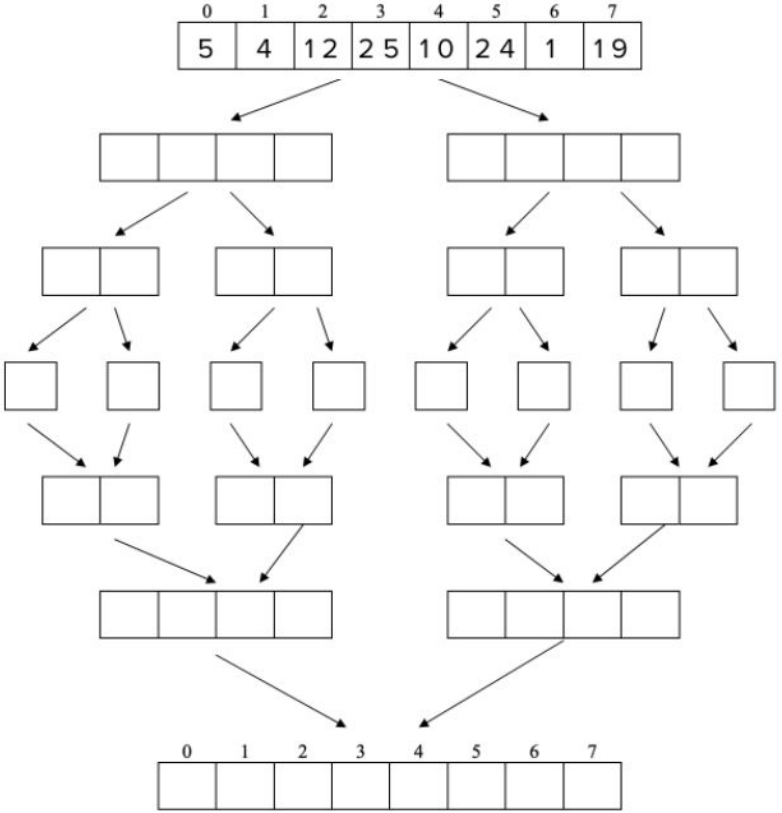
By the end of this lesson, you will be able to . . .

- Compare the efficiency of the merge sort algorithm with the insertion and selection sort algorithms using execution counts
- Identify the benefits and limitations of the merge sort algorithm

🚀 Question of the Day

How does the merge sort algorithm compare in efficiency with the selection and insertion sort algorithms?

A **merge sort** algorithm repeatedly breaks down a list into **sublists** until each sublist consists of a **single element** and merges those **sorted sublists** until it results into a **sorted list**.



The merge sort algorithm uses a **helper function** to perform part of its task.

```
public static void sort(int[] numList, int left, int right) {
  if (left < right) {
    int middle = (left + right) / 2;
    sort(numList, left, middle);
    sort(numList, middle + 1, right);
    merge(numList, left, middle, right);
  }
}
```

A **helper function** is a function that performs a component of another function.

💡 **Discuss:** What are the **base cases**?

```
public static void sort(int[] numList, int left, int right) {
  if (left < right) {
    int middle = (left + right) / 2;
    sort(numList, left, middle);
    sort(numList, middle + 1, right);
    merge(numList, left, middle, right);
  }
}
```

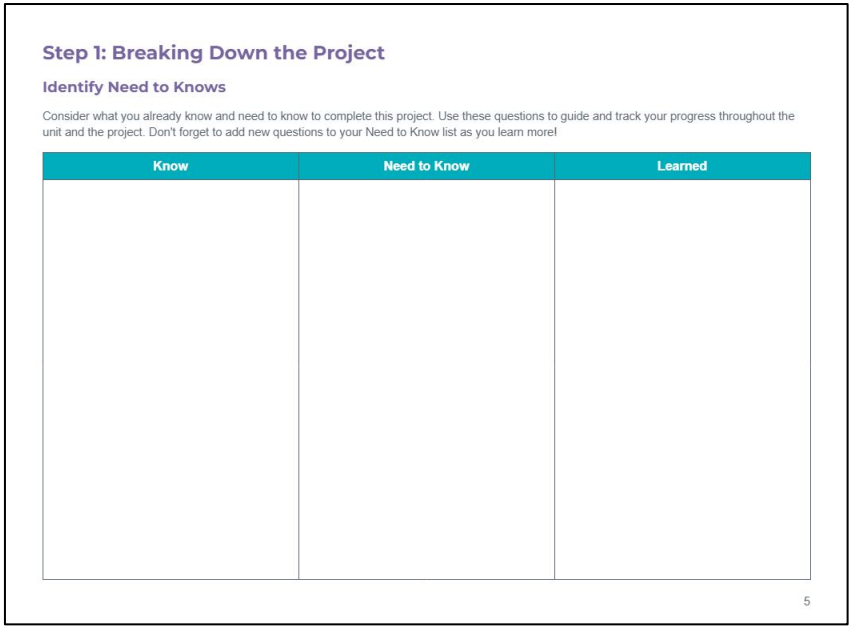
💡 **Discuss:** What is the **recursive call**?

```
public static void sort(int[] numList, int left, int right) {
  if (left < right) {
    int middle = (left + right) / 2;
    sort(numList, left, middle);
    sort(numList, middle + 1, right);
    merge(numList, left, middle, right);
  }
}
```

✅ Do This:

Revisit your **Need to Knows**!

- Check off **answered questions** in the **Need to Know** column.
- Add what you have **learned** and **answers to any questions** in the **Learned** column
- Add any **new questions** to the **Need to Know** column



Step 1: Breaking Down the Project

Identify Need to Knows

Consider what you already know and need to know to complete this project. Use these questions to guide and track your progress throughout the unit and the project. Don't forget to add new questions to your Need to Know list as you learn more!

Know	Need to Know	Learned

5

💼 Key Vocabulary

- **divide-and-conquer:** the process of dividing a problem into smaller problems, solving the smaller problems independently, then combining the solutions to solve the original problem
- **helper function:** the process of dividing a problem into smaller problems, solving the smaller problems independently, then combining the solutions to solve the original problem
- **merge sort:** a divide-and-conquer sorting algorithm that repeatedly breaks down a list into sublists until each sublist consists of a single element and merges those sorted sublists until it results into a sorted list