Lesson Plan: Sorting Race

Year Group: 9 | **Duration:** 50 minutes | **Topic:** Sorting Algorithms

1. Overview

Core Concept: Comparing bubble sort, selection sort, and insertion sort — physically sorting numbered cards while counting comparisons and swaps.

Learning Objectives:

- Execute bubble sort, selection sort, and insertion sort correctly by following written rules
- Count comparisons and swaps for each algorithm on the same data set
- Compare algorithm efficiency and explain why choice of algorithm matters
- Understand that the starting order of data affects performance

Key Vocabulary:

Term	Definition
Sort	Arrange items in a specific order
Comparison	Checking which of two items comes first
Swap	Exchanging the positions of two items
Pass	One complete left-to-right scan through the data
In-place	Sorting without using extra storage
Bubble sort	Repeatedly compare adjacent pairs; swap if out of order
Selection sort	Find the minimum remaining item; place it in position
Insertion sort	Take each item and insert it into its correct position among already-sorted items

2. Before the Lesson

Print:

- ☐ `resource-number-cards.md` — print and cut, 1 set per group (cards 1–10)
- ☐ `resource-algorithm-reference.md` — 1 per group (laminate if possible)
- ☐ `worksheet-tally-sheet.md` — 1 per student

Room Setup: Groups of 3–4.

3. Timed Lesson Flow

0–5 min — Warm-Up Challenge

Give each group a shuffled set of cards. *"Sort them in order. 30 seconds. GO."* Debrief: what method did you use? Was it the same across groups?

5–10 min — Algorithm Overview

Brief introduction of all 3 algorithms. Show the reference card. Teacher acts out each with 5 large cards on the board.

10–20 min — Bubble Sort

Groups sort their shuffled cards using bubble sort ONLY — following the reference card step by step. Count every comparison and every swap.

20–30 min — Selection Sort

Re-shuffle (use the same shuffled starting order if possible). Sort using selection sort. Count comparisons and swaps.

30–40 min — Insertion Sort

Re-shuffle to same starting order. Sort using insertion sort. Count.

40–47 min — Compare Results

Class compiles results on the board. Which algorithm made fewest comparisons? Fewest swaps? Which was easiest to follow?

47–50 min — Discussion: does the starting order matter? (Nearly-sorted data is very efficient for insertion sort. Reverse-sorted data is worst case for bubble sort.)

---

4. Teacher Facilitation Notes

What to look for:

- Students skipping steps or looking ahead — insist on mechanical, step-by-step execution
- Bubble sort: ensure students do COMPLETE passes, not stopping partway through
- Selection sort: ensure students scan ALL remaining cards for each round, not just nearby ones

Common misconceptions:

- Bubble sort is called "bubble" because it's fast — no, slow elements "bubble up" to the top
- Selection sort is the same as insertion sort — they differ fundamentally in approach
- Fewer comparisons always means faster — swaps are also costly; total operations matter

---

5. Extension Tasks

1. Try a nearly-sorted starting order: [1,2,3,4,6,5,7,8,9,10]. Which algorithm handles it most efficiently?
2. Reverse-sorted order: [10,9,8,7,6,5,4,3,2,1]. Which is worst?
3. Research: what is merge sort? How does it differ from the algorithms you tried today?

---

6. Key Takeaway

> **Different sorting algorithms take different numbers of steps — and the best choice depends on your data. Algorithm selection matters as much as algorithm correctness.**