

Exercise walk-through

5.6

Sorting

University Computer Science

Sorting: the inversion bill

- 1 **Inversions** The dataset $[5, 2, 9, 1, 7, 3]$ holds 8 inversions - every adjacent swap fixes exactly one, so swap-adjacent sorts pay 8.
- 2 **Insertion** Shifts per insert: 1, 0, 3, 1, 3 - total 8 shifts and 11 comparisons, result $[1, 2, 3, 5, 7, 9]$.
- 3 **Merge** Halve, sort halves, merge: 10 comparisons on the same data; at $n = 1000$ the families divide by 100.

Insertion-sort snapshots

each row is one insertion; the pale prefix is sorted

start	5	2	9	1	7	3
insert 2 (1 shift)	2	5	9	1	7	3
insert 9 (0 shifts)	2	5	9	1	7	3
insert 1 (3 shifts)	1	2	5	9	7	3
insert 7 (1 shift)	1	2	5	7	9	3
insert 3 (3 shifts)	1	2	3	5	7	9

8 shifts = the inversion count

Insertion sort on the course dataset: each row is one insertion; the pale prefix is already sorted.

Exercises 2.1-2.3

2.1 Count the inversions of $[3, 1, 4, 2]$.

[3]

2.2 Which two of insertion, merge, quick, heap are stable?

[3]

2.3 Sorted input with a first-element pivot is which sort's worst case, and of what class?

[3]

Exercises 2.1-2.3

2.1 Count the inversions of $[3, 1, 4, 2]$. [3]

2.1

- 3 inversions: $(3, 1), (3, 2), (4, 2)$ ✓.

2.2 Which two of insertion, merge, quick, heap are stable? [3]

2.2

- Insertion and merge ✓.

2.3 Sorted input with a first-element pivot is which sort's worst case, and of what class? [3]

2.3

- Quicksort; $O(n^2)$ ✓.

Exercise 3.1

3.1 Sort $[4, 8, 2, 5, 1]$ by insertion sort.

Give the array after each insertion and the shift count of each pass with the running total. Count the dataset's inversions and state the theorem that ties the two totals. Give the comparison count if every pass ends in one failed comparison. [8]

Exercise 3.1

3.1 Sort $[4, 8, 2, 5, 1]$ by insertion sort.

Give the array after each insertion and the shift count of each pass with the running total. Count the dataset's inversions and state the theorem that ties the two totals. Give the comparison count if every pass ends in one failed comparison. [8]

3.1

- Insert 8: $[4, 8]$, 0 shifts ✓.
- Insert 2: $[2, 4, 8]$, 2 shifts ✓.
- Insert 5: $[2, 4, 5, 8]$, 1 shift ✓.
- Insert 1: passes 8, 5, 4, 2 – 4 shifts, giving $[1, 2, 4, 5, 8]$; running total $0 + 2 + 1 + 4 = 7$ ✓.
- Inversions of $[4, 8, 2, 5, 1]$: 7 – equal to the shift total: every shift fixes exactly one inversion ✓.
- Comparisons: 9 – the 7 shift comparisons plus failed ends at 8-vs-4 and 5-vs-4 (the passes that reach the front stop at the boundary, not a comparison) ✓.

Exercise 3.2

3.2 Sort the same $[4, 8, 2, 5, 1]$ by merge sort.

Show the halving tree down to singletons. Merge bottom-up, giving each merge's comparison count and the running total. Compare that total with insertion's on this input, and state each family's worst-case class. [7]

Exercise 3.2

3.2 Sort the same $[4, 8, 2, 5, 1]$ by merge sort.

Show the halving tree down to singletons. Merge bottom-up, giving each merge's comparison count and the running total. Compare that total with insertion's on this input, and state each family's worst-case class. [7]

3.2

- Tree: $[4, 8, 2, 5, 1] \rightarrow [4, 8], [2, 5, 1] \rightarrow [4], [8], [2], [5, 1] \rightarrow [5], [1] \checkmark$.
- Merge $[5], [1]$: 1 comparison $\rightarrow [1, 5] \checkmark$.
- Merge $[2], [1, 5]$: 2 comparisons $\rightarrow [1, 2, 5]$; running 3 $\checkmark$.
- Merge $[4], [8]$: 1; final merge $[4, 8], [1, 2, 5]$: 4 comparisons $\rightarrow [1, 2, 4, 5, 8]$; total $1 + 2 + 1 + 4 = 8 \checkmark$.
- Insertion used 9 comparisons here – merge 8; worst cases: merge $\Theta(n \log n)$, insertion $\Theta(n^2) \checkmark$.