

Exercise walk-through

4.15

Sorting Algorithms

AP Computer Science A

What we will cover

- 1 Ethics of collecting data
- 2 Arrays
- 3 ArrayList
- 4 2D arrays (grids)
- 5 Searching & sorting
- 6 Recursion

4	8	2	9	5
0	1	2	3	4

Arrays

4	8	2	9	5	7
0	1	2	3	4	5

An **array** stores a fixed number of same-type values, reached by **index** (from **0**). Visit every element with a loop:
`for (int i = 0; i < a.length; i++) ...`

Standard algorithms: find the **max/min**, **sum/average**, or count matches by traversing once.



Filing cabinet

Removing from a list while looping over it

the problem

`remove(i)` shifts every later element down, so the element after the removed one is skipped

fix one

loop backwards, from `size()-1` down to 0 – later shifts cannot affect indices you have passed

fix two

do not increment `i` after a removal, so the next pass re-examines the same index

never

remove inside an enhanced for loop – it throws a `ConcurrentModificationException`

the rest

the array algorithms carry over unchanged, plus insertion and deletion, which arrays cannot do easily

This is the single most-tested ArrayList point. If a question removes items in a loop, it is testing exactly this.

Exercises 2.1-2.3

2.1 Describe how selection sort works.

[1]

2.2 Describe how insertion sort works.

[1]

2.3 After one pass of selection sort on {5, 2, 8, 1}, state the array.

[2]

Exercises 2.1-2.3

2.1 Describe how selection sort works.

[1]

2.1

- it repeatedly finds the smallest remaining value and moves it into place at the front of the unsorted part

2.2 Describe how insertion sort works.

[1]

2.2

- it builds a sorted section one element at a time, inserting each new value in order

2.3 After one pass of selection sort on {5, 2, 8, 1}, state the array.

[2]

2.3

- {1, 2, 8, 5} – the smallest (1) is moved to the front

Exercises 3.1-3.2

3.1 Selection sort repeatedly moves the _____ remaining value into place. [1]

☐

- A largest
- B smallest
- C middle
- D random

3.2 Insertion sort builds a [1]

☐

- A random section
- B sorted section one element at a time
- C 2D array
- D linked list

Exercises 3.1-3.2

3.1 Selection sort repeatedly moves the _____ remaining value into place. [1]

- ☐ A largest
☐ B smallest
☐ C middle
☐ D random

3.1 B

- smallest.

3.2 Insertion sort builds a [1]

- ☐ A random section
☐ B sorted section one element at a time
☐ C 2D array
☐ D linked list

3.2 B

- sorted section one element at a time.

Exercise 3.3 (a)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(a) State the array after pass 1.

[1]

Exercise 3.3 (a)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(a) State the array after pass 1.

[1]

(a)

- {1, 4, 3, 2}

Exercise 3.3 (b)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(b) State the array after pass 2.

[1]

Exercise 3.3 (b)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(b) State the array after pass 2.

[1]

(b)

- {1, 2, 3, 4}

Exercise 3.3 (c)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(c) State the fully sorted result.

[1]

Exercise 3.3 (c)

3.3 Apply selection sort to {4, 1, 3, 2} (smallest moved to the front each pass).

(c) State the fully sorted result.

[1]

(c)

- {1, 2, 3, 4}