

Exercise walk-through

4.13

Implementing 2D Array 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 to sum every element of a 2D array. [1]

2.2 State how to process a single row `r` of a 2D array `g`. [1]

2.3 For `int[] [] g = {{1, 2}, {3, 4}};`, find the sum of all elements. [2]

Exercises 2.1-2.3

2.1 Describe how to sum every element of a 2D array. [1]

2.1

- use nested loops to add every element to a running total
-

2.2 State how to process a single row `r` of a 2D array `g`. [1]

2.2

- loop over the elements of `g[r]`
-

2.3 For `int[] [] g = {{1, 2}, {3, 4}};`, find the sum of all elements. [2]

2.3

- $1 + 2 + 3 + 4 = 10$

Exercises 3.1-3.2

3.1 To find the maximum across a whole 2D array, you

[1]

☐

A sum all the elements

B track the largest value over all elements

C sort the rows

D count the elements

☐

3.2 To sum a single column, you fix the

[1]

A row index

B column index

C both indices

D neither index

Exercises 3.1-3.2

3.1 To find the maximum across a whole 2D array, you

[1]

☐

A sum all the elements

B track the largest value over all elements

C sort the rows

D count the elements

3.1 B

- track the largest value over all elements.

3.2 To sum a single column, you fix the

[1]

☐

A row index

B column index

C both indices

D neither index

3.2 B

- column index.

Exercise 3.3 (a)

3.3 `int[] [] g = {{2, 4}, {6, 8}};.`

(a) State the sum of all elements.

[1]

Exercise 3.3 (a)

3.3 `int[] [] g = {{2, 4}, {6, 8}};`

(a) State the sum of all elements.

[1]

(a)

- 20

Exercise 3.3 (b)

3.3 `int[] [] g = {{2, 4}, {6, 8}};.`

(b) State the sum of row 0.

[1]

Exercise 3.3 (b)

3.3 `int[] [] g = {{2, 4}, {6, 8}};.`

(b) State the sum of row 0.

[1]

(b)

- $2 + 4 = 6$

Exercise 3.3 (c)

3.3 `int[] [] g = {{2, 4}, {6, 8}};.`

(c) State the sum of column 1.

[1]

Exercise 3.3 (c)

3.3 `int[] [] g = {{2, 4}, {6, 8}};`

(c) State the sum of column 1.

[1]

(c)

- $4 + 8 = 12$