Grade 9 Computing

Unit 4 · Algorithms and Efficiency

ANSWER KEY · FOR TEACHERS

Full marks 30

Give the marks for the points in the marking scheme; an answer in other words that says the same is right.

1 Key words

1. efficient
2. linear
3. logarithmic
4. quadratic
5. exponential
6. Big-O notation
7. merging
8. deterministic

Each carries 0.5 mark. **K · 4**

2 Multiple choice

i. **b.** 12 **A · 1**

Each doubling adds one look: 2,000 names need 11, 4,000 need 12. (12 looks can search up to 4,095 names.)

ii. **c.** $O(n^2)$ **U · 1**

For each of the n items the inner loop does about n steps: about n × n in all, quadratic.

iii. **c.** 1,024 **A · 1**

Each thing is either in or out, 2 choices, so $2^{10} = 1{,}024$ groups. 3,628,800 is 10 × 9 × … × 1, the number of orders the 10 things can be put in, which grows even faster.

iv. **c.** 200 seconds **A · 1**

Ten times the items makes a quadratic algorithm's work 10 × 10 = 100 times as much: 100 × 2 = 200 seconds, more than 3 minutes. A linear one would take 20 seconds.

v. **b.** the number of comparisons it makes **U · 1**

The pivots are chosen by chance, so the piles, the rounds and the comparisons differ. But whatever the pivots, the piles always end in order, so the sorted list is the same every time.

3 What does it print?

MARKING SCHEME **U · 3**

- ✓ `4 6` and `8 28`: the inner loop starts at `i + 1`, so each pair is counted once.
- ✓ `16 120`.
- ✓ Quadratic: doubling n makes about 4 times the steps (a loop inside a loop).

4 Short answers

a. **MARKING SCHEME** **A · 2**

- ✓ 30 × 29 ÷ 2 = 435 matches.
- ✓ 60 × 59 ÷ 2 = 1,770 matches, about 4 times as many (quadratic).

b. **MARKING SCHEME** **A · 2**

- ✓ Quadratic: 3 times the records, 3 × 3 = 9 times the work, about 27 seconds.
- ✓ Linear: 3 times the work, about 9 seconds.

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

c. MARKING SCHEME U · 2

- ✓ A deterministic algorithm takes the same steps and gives the same output every time for the same input; a random one makes some choices by chance, so its steps (and sometimes its answer) can differ from run to run.
- ✓ An example of each: deterministic, such as binary search or merge sort; random, such as quicksort with a random pivot, estimating π with random dots, or shuffling cards.

5 Trace the program

ANSWER A · 4

a[i]	b[j]	Added to c	i	j
(before)			0	0
12	8	8	0	1
12	31	12	1	1
30	31	30	2	1
45	31	31	2	2
45	50	45	3	2
66	50	50	3	3

```
OUTPUT
[8, 12, 30, 31, 45, 50, 66]
```

The loop stops when `j` reaches 3, the end of `b`; line 12 then puts 66, the rest of `a`, on the end. 6 comparisons for 7 numbers.

MARKING SCHEME

- ✓ The first three rows: 8, then 12, then 30 added, with `j` 1 and `i` 1, then 2.
- ✓ The next three: 31, 45 and 50 added, ending with `i` 3 and `j` 3.
- ✓ The loop stops after 6 rows, because `j` is 3, the length of `b`.
- ✓ Output `[8, 12, 30, 31, 45, 50, 66]`: line 12 puts 66 on the end.

6 Find the bug

MARKING SCHEME A · 2

- ✓ `[8, 12, 30, 31, 45, 50]`, without 66.
- ✓ The loop ends when one list is used up, and line 12 is what puts the rest of the other list on the end.

7 Long answer

ANSWER

```
ALGORITHM
Step 1: Start
Step 2: Read the list of n numbers
Step 3: Set i = 1
Step 4: Is i < n?
        Yes: go to Step 5
        No: Display "in order", go to Step 7
Step 5: Is number i > number i + 1?
        Yes: Display "not in order", go to Step 7
        No: go to Step 6
Step 6: Set i = i + 1, go to Step 4
Step 7: Stop
```


Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

It compares each pair of neighbors once, so at most n − 1 comparisons (when the list is in order, or only the last pair is wrong). That is linear growth, O(n). (Here the numbers are counted from 1, as in everyday speech; in Python they would be positions 0 to n − 1.) A · 4

MARKING SCHEME

- ✓ Start, read the list, and a counter set to the first position.
- ✓ A loop that compares each number with the next one (number i > number i + 1?), moving on by one each time.
- ✓ Displays "not in order" and stops as soon as a pair is the wrong way round; "in order" when the end is reached; then Stop.
- ✓ At most n − 1 comparisons: linear growth, O(n).

8 Think

ANSWER HA · 2

Only about 3 times bigger. A quadratic sort on a list 3.16 times as long does 3.16 × 3.16 ≈ 10 times the work, which the 10 times faster computer does in the same time. With an O(n log n) sort, the same computer could handle almost 10 times as many items. And switching the old computer to merge sort would do far more for big lists than the new computer: for 100,000 items it does about 3,000 times less work. A better algorithm usually beats a faster computer.

MARKING SCHEME

- ✓ Only about 3 times bigger: a list 3.16 times as long gives a quadratic sort 3.16 × 3.16 ≈ 10 times the work.
- ✓ A better algorithm (an O(n log n) sort such as merge sort) would help far more than a faster computer for big lists.

9 Activity: Watch the sorts

(not marked)

WHAT STUDENTS SHOULD FIND

- **Mixed:** merge sort 17 comparisons, shorter passes 27, insertion sort 19.
- **Nearly in order:** merge sort 13, shorter passes 13, insertion sort 8.
- **Backwards:** merge sort 12, shorter passes 28, insertion sort 28.

Merge sort does the least work on the mixed and backwards lists; on a list nearly in order, insertion sort wins. For 8 items the differences are small; for big lists merge sort wins by far.

Marks by level	Knowledge	Understanding	Application	Higher ability	Total
Marks	4	7	17	2	30

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking