Grade 13 A/L ICT · Lesson 9

9.13 Searching and Sorting: Sequential Search and Bubble Sort

Syllabus 9.13. Sequential (linear) search checks items one by one from the start until the target is found or the list ends. **Bubble sort** compares neighbouring items and swaps them if they are in the wrong order; after each **pass** the largest remaining value “bubbles” to the end.

1. Sequential search

🐍 Sequential search

```
def linear_search(items, target):
    for i in range(len(items)):
        if items[i] == target:
            return i
    return -1

data = [34, 7, 23, 32, 5, 62]
print(linear_search(data, 23))
print(linear_search(data, 99))
```

▶ Output

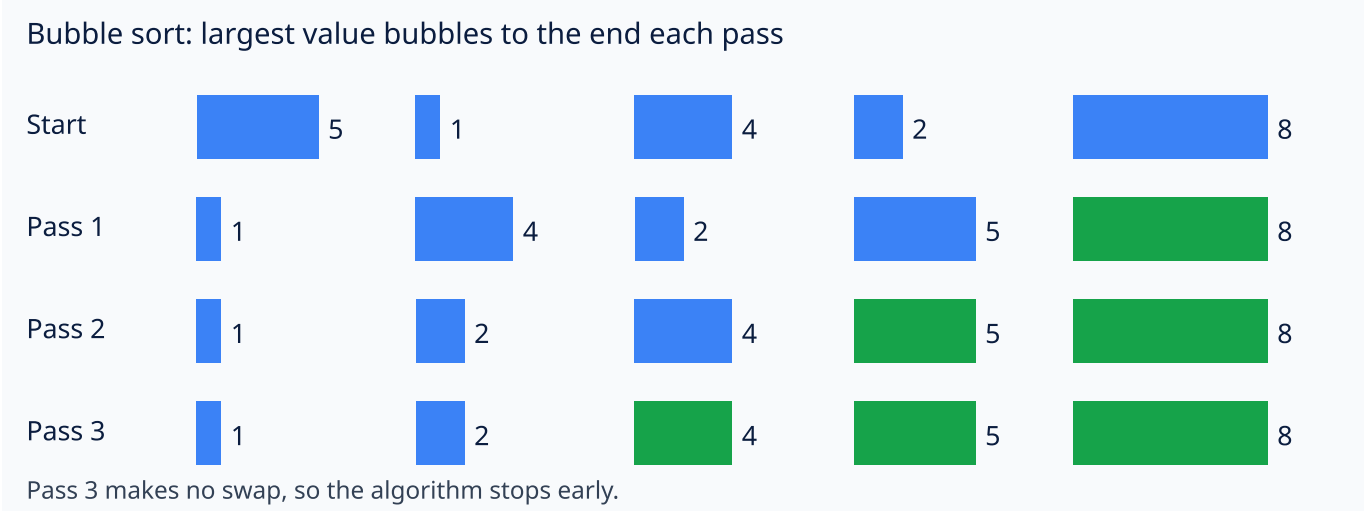
```
2
-1
```

Target	Result index	Comparisons
23	2	3
99	−1	6

🧱 **Worked example: How many comparisons to find 23 and 99 in [34, 7, 23, 32, 5, 62]?**
23 is checked against 34, 7, 23 – found at index **2** after **3** comparisons. 99 is compared with all **6** items and the function returns **−1**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Bubble sort



Pass 1 of bubble sort

🐍 Bubble sort showing each pass

```
def bubble_sort(a):
    n = len(a)
    for p in range(n - 1):
        swapped = False
        for j in range(n - 1 - p):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
                swapped = True
        print("Pass", p + 1, a)
        if not swapped:
            break
    return a

bubble_sort([5, 1, 4, 2, 8])
```

▶ Output

```
Pass 1 [1, 4, 2, 5, 8]
Pass 2 [1, 2, 4, 5, 8]
Pass 3 [1, 2, 4, 5, 8]
```

Pass	List after the pass
1	[1, 4, 2, 5, 8]
2	[1, 2, 4, 5, 8]
3	[1, 2, 4, 5, 8]

🧱 Worked example: Sort [5, 1, 4, 2, 8] with bubble sort.
Pass 1 swaps 5↔1, 5↔4, 5↔2 → [1, 4, 2, 5, 8]. Pass 2 swaps 4↔2 → [1, 2, 4, 5, 8]. Pass 3 makes no swap, so the swapped flag stops the sort early. Total swaps: **4**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

💡 For n items bubble sort needs at most n − 1 passes; sequential search needs at most n comparisons.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Python's built-in sorted() uses Timsort, much faster than bubble sort on large data – but bubble sort is the classic for learning how sorting works.

Key Words

Searching – Finding whether and where a value is in a list.	**Sequential search** – Checking each item in turn from the start.
Target – Value being searched for.	**Comparison** – Checking one item against the target or a neighbour.
Sorting – Arranging items in ascending or descending order.	**Bubble sort** – Sort that swaps neighbouring items in the wrong order.
Pass – One full run through the list in bubble sort.	**Swap** – Exchanging the positions of two items.
Swapped flag – Boolean that stops the sort when a pass has no swaps.	**Ascending order** – Smallest to largest.
Return −1 – Signal that the target was not found.	**Algorithm efficiency** – How the work grows as the data grows.

Sample Q&A

1. How does sequential search work?

> **Answer:** Checks items one by one until found or the end.

2. Index of 23?

> **Answer:** 2.

3. Comparisons for 99?

> **Answer:** 6 (all items), returns −1.

4. List after pass 1?

> **Answer:** [1, 4, 2, 5, 8].

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Why does it stop after pass 3?

> **Answer:** No swaps, so swapped stays False.

6. Maximum passes for 5 items?

> **Answer:** 4 (n − 1).

MCQ

1. Sequential search for 23 in [34, 7, 23, 32, 5, 62] returns

A. 2
B. −1
C. 3
D. 23
E. 1

2. Comparisons needed to find 23 are

A. 2
B. 1
C. 4
D. 3
E. 6

3. Searching for 99 returns

A. 6
B. 99
C. −1
D. None
E. 0

4. After pass 1, [5, 1, 4, 2, 8] becomes

A. [5, 4, 2, 1, 8]
B. [1, 4, 2, 5, 8]
C. [8, 5, 4, 2, 1]
D. [1, 2, 4, 5, 8]
E. [1, 5, 4, 2, 8]

5. Total swaps sorting [5, 1, 4, 2, 8] are

A. 4
B. 3
C. 5
D. 6
E. 2

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. Bubble sort compares

A. random pairs
B. every item with the target
C. the middle item
D. first and last only
E. neighbouring items

7. The swapped flag lets bubble sort

A. swap faster
B. skip pass 1
C. search for a value
D. stop early when a pass has no swaps
E. sort descending

8. The maximum passes for n items is

A. n + 1
B. n^2
C. n − 1
D. 2n
E. n

9. Which term means: “Finding whether and where a value is in a list”?

A. Bubble sort
B. Return −1
C. Swap
D. Target
E. Searching

10. Which term means: “Checking each item in turn from the start”?

A. Pass
B. Sequential search
C. Swapped flag
D. Comparison
E. Algorithm efficiency

11. Which term means: “Value being searched for”?

A. Target
B. Searching
C. Ascending order
D. Swap
E. Sorting

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

12. Which term means: “Checking one item against the target or a neighbour”?

A. Swapped flag
B. Return −1
C. Comparison
D. Bubble sort
E. Sequential search

13. Which term means: “Arranging items in ascending or descending order”?

A. Ascending order
B. Sorting
C. Target
D. Algorithm efficiency
E. Pass

14. Which term means: “Sort that swaps neighbouring items in the wrong order”?

A. Return −1
B. Bubble sort
C. Comparison
D. Swap
E. Searching

15. Which term means: “One full run through the list in bubble sort”?

A. Algorithm efficiency
B. Sequential search
C. Pass
D. Swapped flag
E. Sorting

16. Which term means: “Exchanging the positions of two items”?

A. Swap
B. Bubble sort
C. Searching
D. Target
E. Ascending order

17. Which term means: “Boolean that stops the sort when a pass has no swaps”?

A. Sequential search
B. Return −1
C. Pass
D. Comparison
E. Swapped flag

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

18. Which term means: “Smallest to largest”?

A. Algorithm efficiency
B. Sorting
C. Target
D. Ascending order
E. Swap

19. Which term means: “Signal that the target was not found”?

A. Comparison
B. Swapped flag
C. Searching
D. Bubble sort
E. Return −1

20. Which term means: “How the work grows as the data grows”?

A. Sequential search
B. Ascending order
C. Sorting
D. Algorithm efficiency
E. Pass

Structured Questions

1. Write a Python function for sequential search. *(3 marks)*

> for i in range(len); if equal return i; return −1.

2. Find 23 and 99 in [34, 7, 23, 32, 5, 62] with comparison counts. *(2 marks)*

> 23: index 2, 3 comparisons; 99: −1, 6 comparisons.

3. Show each pass of bubble sort on [5, 1, 4, 2, 8]. *(3 marks)*

> [1, 4, 2, 5, 8]; [1, 2, 4, 5, 8]; no swaps, stop.

4. Explain the purpose of the swapped flag. *(3 marks)*

> Detects a pass with no swaps; ends early; saves time.

5. Compare the best and worst case of sequential search. *(2 marks)*

> Best 1 comparison (first item); worst n (last or missing).

Summary

- Sequential search checks items in order.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

- It returns the index or −1.
- Bubble sort swaps neighbours in the wrong order.
- Each pass puts the largest left value at the end.
- A pass with no swaps ends the sort.

MCQ answer key – 9.13

1. **A** – Index from 0.
2. **D** – 34, 7, 23.
3. **C** – Not found.
4. **B** – 5 bubbles to its place.
5. **A** – 3 + 1.
6. **E** – Adjacent.
7. **D** – Early exit.
8. **C** – n − 1.
9. **E** – Searching – Finding whether and where a value is in a list.
10. **B** – Sequential search – Checking each item in turn from the start.
11. **A** – Target – Value being searched for.
12. **C** – Comparison – Checking one item against the target or a neighbour.
13. **B** – Sorting – Arranging items in ascending or descending order.
14. **B** – Bubble sort – Sort that swaps neighbouring items in the wrong order.
15. **C** – Pass – One full run through the list in bubble sort.
16. **A** – Swap – Exchanging the positions of two items.
17. **E** – Swapped flag – Boolean that stops the sort when a pass has no swaps.
18. **D** – Ascending order – Smallest to largest.
19. **E** – Return −1 – Signal that the target was not found.
20. **D** – Algorithm efficiency – How the work grows as the data grows.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.