



F5.7 Bubble sort

Algorithms · GCSE · OCR J277 2.1.3, AQA 8525 3.1.4, Edexcel 1CP2 1.2.6 ·
about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Passes and swaps, stopping early, and hearing a tune become a scale.

- The algorithm
- In Python
- Hear it sort
- How much work it does

Questions

5 marks in all

1. What does this program print?

[1 mark]

```
items = [5, 2, 4, 1]
for i in range(len(items) - 1):
    if items[i] > items[i + 1]:
        items[i], items[i + 1] = items[i + 1], items[i]
print(items)
```

2. When can bubble sort stop early?

[1 mark]

- ☐ A After a whole pass with no swaps
- ☐ B After the first swap
- ☐ C When it reaches the middle of the list
- ☐ D It can never stop early

3. What does bubble sort compare?

[1 mark]

- ☐ A Each pair of neighbouring items
- ☐ B Each item with the first item
- ☐ C The middle item with the target
- ☐ D Two halves of the list

4. A list of 6 items. How many comparisons does one pass of bubble sort make?

[1 mark]

5. Why does a swap in pseudocode usually need a temporary variable?

[1 mark]

- ☐ A Copying one item over the other would lose the first value
- ☐ B Pseudocode has no lists
- ☐ C It makes the sort faster
- ☐ D Python requires it

The task: sort the tune

Sort `notes = [392, 262, 523, 330, 294, 440, 349, 494]` with **bubble sort**, written yourself (no `sort` or `sorted`). Then print `sorted: <the list>` and play every note of the sorted list for 0.2 seconds.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
```

```
notes = [392, 262, 523, 330, 294, 440, 349, 494]
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f5-7-bubble-sort/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Count the comparisons and swaps, and print both at the end. Try a list that is already sorted, and one in reverse order.
2. Change the sort to put the notes in descending order: a falling tune.
3. Each pass can stop one item earlier than the last, because the end of the list is already sorted. Add that improvement and count the comparisons saved.