



F5.8 Insertion sort

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

What this lesson is about

Building a sorted part one item at a time, on the robot's readings.

- The algorithm
- In Python
- Sorting the robot's readings
- Bubble sort or insertion sort?

Questions

4 marks in all

1. How does insertion sort work?

[1 mark]

- ☐ A It takes each item in turn and inserts it into its place in a growing sorted part
- ☐ B It compares neighbours and swaps them
- ☐ C It splits the list and merges it
- ☐ D It checks the middle item

Answer: A. Like sorting a hand of cards one card at a time.

2. What does this program print?

[1 mark]

```
items = [31, 15, 51, 20]
for i in range(1, len(items)):
    current = items[i]
    j = i - 1
    while j >= 0 and items[j] > current:
        items[j + 1] = items[j]
        j = j - 1
    items[j + 1] = current
print(items)
```

Answer:

[15, 20, 31, 51]

Each item is shifted left past bigger ones: the result is in ascending order.

3. On which kind of list is insertion sort especially quick?

[1 mark]

- ☐ A One that is nearly sorted
- ☐ B One in reverse order
- ☐ C One with many repeated items
- ☐ D A very large random list

Answer: A. Each item only moves a short way, so there is little shifting.

4. In insertion sort of [31, 15, 51, 20], what does the list look like after the first item is inserted?

[1 mark]

- ☐ A [15, 31, 51, 20]
- ☐ B [31, 15, 51, 20]
- ☐ C [15, 20, 31, 51]
- ☐ D [15, 31, 20, 51]

Answer: A. 15 is taken and moved in front of 31. The rest have not been touched yet.

The task: sort the survey

Look in eight directions, 45 degrees apart, recording `distance()` in a list. Sort the list with **insertion sort**, written yourself (no `sort` or `sorted`), printing the list after each insertion. Finally print `sorted: <the list>`.

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

readings = []
for i in range(8):
    readings.append(distance())
    turn_right(30, angle=45)
```

The hint students can ask for: Take each item in turn as the one to place, and shift the sorted items on its left along until there is a gap for it. Print the list after each item is placed.

A solution

```
from bugbot import *
connect()
readings = []
for i in range(8):
    readings.append(distance())
    turn_right(30, angle=45)

for i in range(1, len(readings)):
    current = readings[i]
    j = i - 1
    while j >= 0 and readings[j] > current:
        readings[j + 1] = readings[j]
        j = j - 1
    readings[j + 1] = current
    print("after inserting", current, ":", readings)
print("sorted:", readings)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.