



F5.8 Insertion 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

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

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)
```

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

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]

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)
```

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



Do it on the robot

www.bugbotlab.com/learn/f5-8-insertion-sort/

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

Challenges

1. Count how many items were shifted in total. Try it on a list that is already sorted, and one in reverse order.
2. Sort a list of names alphabetically with insertion sort.
3. Keep a list sorted as you build it: insert each new reading into its place as soon as it is measured, instead of sorting at the end.