



A5.1 Comparing algorithms

Algorithms and complexity · A level · OCR H446 2.3.1, AQA 7517 4.4.4.1,
Eduqas A500QS 1.3 · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

Time and space efficiency as functions of the size of the problem; linear, polynomial, exponential and logarithmic functions; permutations and $n!$.

- Why not use a stopwatch?
- Time and space
- The maths: four kinds of function
- Permutations

Questions

6 marks in all

- Why are algorithms compared by counting operations as a function of n rather than by timing them? [1 mark]
☐ A A timing depends on the computer, the language and the data, but the growth in operations does not
☐ B Timing a program is not possible on modern computers
☐ C Counting operations always gives a smaller number
☐ D Timings are only accurate for sorting algorithms
- Which of these is an exponential function of n ? [1 mark]
☐ A 2^n
☐ B n^2
☐ C $2n$
☐ D $\log_2 n$
- What is $\log_2 256$? [1 mark]

- In how many different orders can a robot visit 5 distinct checkpoints? [1 mark]

- Checking a list for repeats by storing every item seen in a set, instead of comparing every pair, makes which of these true? [1 mark]

Tick every answer that is true.
☐ A It makes fewer comparisons
☐ B It uses more memory
☐ C It uses less memory
☐ D It needs the list to be sorted

6. What does this print?

[1 mark]

```
n = 1
for k in range(1, 6):
    n = n * k
print(n, 2 ** 5, 5 ** 2)
```

The task: the growth table

Print a table of how four functions grow. For each n in 4, 8, 16, 32 and 64, print one line in exactly this form: `n=8 log2=3 square=64 exponential=256` where `log2` is $\log_2 n$ (a whole number here, because every n is a power of 2), `square` is n^2 and `exponential` is 2^n . Work out `log2` by counting how many times n can be halved (with `// 2`) before it reaches 1, not with the `math` module.

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

n = 8
print("n=" + str(n))
```

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



Do it on the robot

www.bugbotlab.com/learn/a5-1-comparing-algorithms/

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

Challenges

1. From which n onwards is 2^n always bigger than n^2 ? Check your answer with a loop up to $n = 30$.
2. The robot's memory holds 64,000 IDs. Which duplicate check from this lesson could it run on 50,000 IDs, and which could it not? Explain.
3. How many different orders are there for 5 checkpoints if the robot must always start at A?