

3.17 Algorithmic Efficiency

Name: _____ Class: _____ Date: _____

Total: 8 marks

Objective

Build the skills to answer exam questions on **algorithmic efficiency**.

You must be able to:

- explain how **algorithmic efficiency** 算法效率 compares algorithms by the resources they use
- estimate the number of **steps** an algorithm takes as the input grows
- distinguish a **reasonable** from an **unreasonable** running time
- use a **heuristic** 启发式 when an exact solution is too slow

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Efficiency

Compares algorithms by the **resources** (steps or time) they use as the input grows.

■ Reasonable vs unreasonable

A **reasonable** running time grows slowly (e.g. proportional to n or n^2). An **unreasonable** one grows explosively (e.g. 2^n), becoming impractical on large inputs.

■ Heuristics

When an exact solution is too slow, a **heuristic** finds a **good-enough** answer quickly.

2 Practice

2.1 State what algorithmic efficiency compares. [1]

2.2 State what a heuristic is. [1]

2.3 State why some correct algorithms are not used in practice. [1]

3 Exam-style questions

3.1 Algorithmic efficiency compares algorithms by [1]

- **A** their length in lines
 - **B** the resources they use
 - **C** their author
 - **D** the language used
-

3.2 A heuristic gives a [1]

- **A** guaranteed exact answer
 - **B** good-enough answer quickly
 - **C** deliberately wrong answer
 - **D** random answer
-

3.3 Algorithm A takes about n steps; algorithm B takes about 2^n steps.

- (a) State which is more efficient for large n . [1]
- (b) State whether B could be too slow to use. [1]
- (c) State one option when an exact solution is too slow. [1]

4 Go further

- work through the **3.17 Algorithmic Efficiency** lesson on the **Learn** page;
- read the **Algorithms and Programming** section of the AP Computer Science Principles handout on the **Know** page.

Solutions

2.1 the resources (steps or time) that algorithms use.

2.2 an approach that finds a good-enough solution when an exact one is too slow.

2.3 they take an unreasonable (too long) time on large inputs.

3.1 B.

3.2 B.

3.3 (a) A. (b) yes. (c) use a heuristic.