

3.18 Undecidable Problems

Name: _____ Class: _____ Date: _____

Total: 8 marks

Objective

Build the skills to answer exam questions on **undecidable problems**.

You must be able to:

- explain how a **decidable** 可判定 problem has an algorithm that always gives a correct yes-or-no answer
- explain how an **undecidable** 不可判定 problem has no algorithm that solves every case
- describe the difference between undecidable and merely **inefficient** 低效
- recognize undecidability as a **limit** on computation

1 Worked examples

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

■ Decidable vs undecidable

- A **decidable** problem has an algorithm that **always** gives a correct yes-or-no answer.
- An **undecidable** problem has **no** algorithm that solves **every** case correctly.

■ Undecidable is not just slow

Undecidability is stronger than inefficiency: an inefficient problem can still be solved (given enough time), but an undecidable one **cannot** be solved for all cases at all.

■ A limit on computation

Undecidability shows there are things computers simply **cannot** do.

2 Practice

2.1 Define a decidable problem.

[1]

2.2 Define an undecidable problem.

[1]

2.3 State the difference between an undecidable problem and a merely inefficient one. [1]

3 Exam-style questions

3.1 A decidable problem has an algorithm that [1]

- **A** always gives a correct yes-or-no answer
 - **B** never finishes
 - **C** gives a random answer
 - **D** is simply slow
-

3.2 An undecidable problem [1]

- **A** can always be solved slowly
 - **B** has no algorithm that solves every case correctly
 - **C** is a syntax error
 - **D** is a data type
-

3.3 A problem has no algorithm that can correctly answer it in **all** cases.

(a) Name this kind of problem. [1]

(b) State whether this is the same as being slow. [1]

(c) State what it shows about computation. [1]

4 Go further

- work through the **3.18 Undecidable Problems** 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 a problem with an algorithm that always gives a correct yes-or-no answer.

2.2 a problem for which no algorithm can solve every case correctly.

2.3 an inefficient problem can still be solved (just slowly); an undecidable one cannot be solved for all cases at all.

3.1 A.

3.2 B.

3.3 (a) an undecidable problem. (b) no. (c) that there are limits to what computers can do.