AP COMPUTER SCIENCE A • EXERCISE SHEET

4.17 Recursive Searching and Sorting

Name: _____ Class: _____ Date: _____ **Total: 9 marks**

Objective

Build the skills to answer exam questions on **recursive searching and sorting**.

You must be able to:

- describe how **binary search** can be written as a **recursive** method
- trace a recursive binary search halving the range on each call
- describe how **merge sort** 归并排序 splits, sorts, and merges

1 Worked examples

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

Recursive binary search

Binary search is naturally recursive: check the **middle**; if the target is smaller, **recurse on the left half**, else recurse on the right —each call halves the range.

Merge sort

1. **Split** the collection into two halves.
2. **Sort each half** (recursively, using merge sort).
3. **Merge** the two sorted halves into one sorted collection.

2 Practice

2.1 State how binary search can be written recursively. [1]

2.2 State the three steps of merge sort. [2]

2.3 State what merge sort does after sorting the two halves. [1]

3 Exam-style questions

3.1 Merge sort works by [1]

- **A** checking each element in turn
- **B** splitting, sorting each half, and merging
- **C** swapping neighbouring elements only
- **D** not sorting at all

3.2 A recursive binary search calls itself on [1]

- **A** the whole list again
- **B** the correct half of the list
- **C** a random half
- **D** nothing

3.3 Merge sort is applied to a list.

(a) State the first step. [1]

(b) State what happens to each half. [1]

(c) State the final step. [1]

4 Go further

- work through the **4.17 Recursive Searching and Sorting** lesson on the **Learn** page;
- read the **Recursion** section of the AP Computer Science A handout on the **Know** page.

Solutions

2.1 it checks the middle element, then calls itself on the half that could contain the target.

2.2 split into two halves; sort each half; merge them.

2.3 it merges the two sorted halves into one sorted list.

3.1 B.

3.2 B.

3.3 (a) split the list into two halves. (b) each half is sorted (recursively). (c) merge the two sorted halves.