

A merge sort is carried out on this list.

9 4 3 5 5 1 7

(i) State the number of splits required to complete the sort.

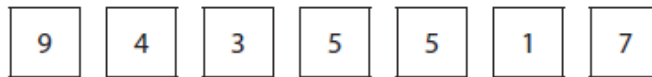
(1)

.....

(ii) Here is the list after being split.

Complete the diagram to show the merging steps.

(2)



(iii) Explain the effect on efficiency of using a merge sort algorithm instead of a bubble sort algorithm on the original list.

9 4 3 5 5 1 7

(2)

ANSWERS

1. Merge sort is a **recursive divide and conquer** algorithm that repeatedly **divides the array in half** until it cannot be divided any further. At that point, each subarray contains only one element.

In this case, the initial list is split into smaller subarrays as follows:

First split: [9, 4, 3, 5] | [5, 1, 7]

Second split: [9, 4] | [3, 5] | [5, 1] | [7]

Third split: [9] | 4 | 3 | [5] | [5] | 1 | [7]

So the answer is 3 splits.

2. The sorted subarrays are then **merged** back together to form a single sorted array. The merging process compares elements from each subarray and combines them in order. For example, merging [9] and 4 results in [4, 9].

The final merge combines all the subarrays to create the fully sorted list:

- a. [4, 9] | [3, 5] | [1, 5] | [7]
- b. [3, 4, 5, 9] | [1, 5, 7]
- c. [1, 3, 4, 5, 5, 7, 9]

3. **When discussing efficiency in algorithms talk about the speed it takes for the algorithm to complete and the memory that the algorithm uses.**

Bubble sort is a simple comparison-based sorting algorithm.

It repeatedly compares adjacent elements in the list and swaps them if they are in the wrong order.

The process continues until the entire array is sorted. Bubble sort becomes **less efficient** as data sets grow whereas merge sort becomes **more efficient** as data sets increase in size.

Bubble sort uses less memory because it sorts-in-place whereas merge sort creates new arrays that need to be stored in memory. However, the time advantage of merge sort outweighs the additional use of memory.