# AQA Computer Science GCSE
## 3.1.2 Efficiency of Algorithms

Flashcards

# What does algorithm efficiency refer to?

# What does algorithm efficiency refer to?

How well an algorithm performs, especially in terms of the time it takes to complete a task.

# Can more than one algorithm solve the same problem?

Can more than one algorithm solve the same problem?

Yes, different algorithms (e.g., bubble sort vs. merge sort) can solve the same problem but vary in speed, efficiency, and suitability.

# What factors affect an algorithm's efficiency?

# What factors affect an algorithm's efficiency?

The size of the data set, the nature of the data, and the algorithm's design.

# What's the key idea when choosing between algorithms?

# What's the key idea when choosing between algorithms?

Just because an algorithm works doesn't mean it's the best choice.

Some algorithms also may not work on certain types of data, e.g. binary search does not work on unsorted lists.

# What does the spec require when comparing algorithms?

# What does the spec require when comparing algorithms?

Compare time efficiency in simple terms—no need for Big O notation.

# What does time efficiency involve?

# What does time efficiency involve?

The number of steps and time taken to complete a task, especially as input size increases.

# Which is better for unsorted data: Linear or Binary Search?

# Which is better for unsorted data: Linear or Binary Search?

Linear Search—it doesn't require sorted data.

# Which is faster on large data sets: Bubble Sort or Merge Sort?

Which is faster on large data sets: Bubble Sort or Merge Sort?

Merge Sort is generally much faster on large data sets because of its divide and conquer approach.

# What makes one algorithm more efficient than another?

What makes one algorithm more efficient than another?

Fewer steps, better performance on large data, and avoiding unnecessary operations.

# What is a simple method to compare two algorithms?

# What is a simple method to compare two algorithms?

Look at how many steps they take and how well they scale with larger input sizes.

Give an example where Binary Search is more efficient.

Give an example where Binary Search is more efficient.

When searching a sorted list—Binary Search is much faster than Linear Search.