Linear search: time complexity

Challenge 1

C C

Which of the following is the worst-case time complexity of a linear search, expressed in Big O notation?

- $\mathcal{O}(\log n)$
- $\mathcal{O}(n)$
- $\mathcal{O}(n^2)$
- $\mathcal{O}(1)$

Binary search: time complexity

A Level Advanced
C C C C

Which of the following is the worst-case time complexity of a binary search, expressed in Big O notation?

- ○ $\mathcal{O}(n^2)$
- ○ $\mathcal{O}(\log n)$
- ○ $\mathcal{O}(n)$
- ○ $\mathcal{O}(n \log n)$

Search complexity

A Level Advanced

C C C C

Bronwin wants to create a program that will search an array of student names and return `True` if a given student has volunteered for the Year 7 Open Evening. She is considering whether to use a **linear search** or a **binary search** for this purpose.

Each item of the array is a **record with three fields**:

- `ID` : an ID that increments,
- `student_name` : the name of the student,
- `volunteer` : a boolean variable that specifies if they agreed to volunteer or not.

The first three elements of the array are:

```
[['ST201', 'Simon Brown', True], ['ST202', 'Kate Morris', False], ['ST203',
'Alicia Huston', False]]
```

In relation to the use of a linear search or binary search to find an item in an array of records, which of the following statements is **correct**?

○ Binary search will find the student much faster than linear search if the array is searched by using the `student_name` field.

○ Binary search can find a student in maximum $logn + 1$ comparisons (where n is the number of elements in the array) if the `ID` is used to search for a student.

○ Linear search is the only algorithm that is suitable for finding a student in the array if the student list is below 100 entries.

○ Linear search has a time complexity of $O(n^2)$ (where n is the number of elements in the array) as it needs to go through the array and check all three fields for each student.

Bubble sort: time complexity

A Level Advanced

C C C C

Which of the following is the time complexity of a bubble sort?

- ○ $\mathcal{O}(n \log n)$
- ○ $\mathcal{O}(n)$
- ○ $\mathcal{O}(n^2)$
- ○ $\mathcal{O}(\log n)$

Insertion sort: time complexity

Challenge 1

C C

Which of the following is the worst-case time complexity of an insertion sort, expressed in Big O notation?

- ○ $\mathcal{O}(\log n)$
- ○ $\mathcal{O}(n^2)$
- ○ $\mathcal{O}(2^n)$
- ○ $\mathcal{O}(n)$

Merge sort: time complexity

A Level Advanced
C C C C

Which of the following is the worst-case time complexity of a merge sort, expressed in Big O notation?

- ○ $\mathcal{O}(n)$
- ○ $\mathcal{O}(n^2)$
- ○ $\mathcal{O}(n \log n)$
- ○ $\mathcal{O}(\log n)$

Quick sort: time complexity

Challenge 1

C C

Which of the following is the worst-case time complexity of a quick sort, expressed in Big O notation?

- $\mathcal{O}(n \log n)$
- $\mathcal{O}(n)$
- $\mathcal{O}(n^2)$
- $\mathcal{O}(\log n)$

Raspberry Pi Foundation

Compare sorting algorithms

A Level Advanced
C C C C

Your classmate asks you for help in picking an efficient sorting method to use in his zoo application.

He wants to sort the list of animals below into alphabetical order:

animals = ['aoudad', 'camel', 'cheetah', 'crow', 'baboon', 'deer', 'hare', 'leopard', 'mink', 'peccary', 'moose', 'mule', 'parrot']

Which of the below statements is correct?

- ○ Bubble sort will make the fewest comparisons because it stops as soon as the list is sorted.
- ○ Insertion sort will make the fewest comparisons because the list is almost sorted.
- ○ Bubble and insertion sort will take the same amount of time to sort the list because they both have a time complexity of $O(n^2)$.
- ○ Merge sort will be the most efficient choice because it has the lowest time and space complexity.

Quick vs bubble sort

A Level Advanced
C C C C

Two sorting algorithms are quick sort and bubble sort. Two students have been arguing about which is the best to use. Barack favours the quick sort algorithm whereas Salome is championing the bubble sort.

Both students have written two statements in support of their favoured algorithm but only one of the statements is correct. Which one is it?

- ○ The bubble sort algorithm has lower space complexity than quick sort.
- ○ The quick sort algorithm is better than the bubble sort algorithm for sorting a set of data that is already substantially in order.
- ○ The quick sort algorithm is more suitable for sorting large data sets than the bubble sort algorithm.
- ○ On average, the bubble sort algorithm is faster at sorting data than the quick sort algorithm.

Quick vs merge sort

A Level Advanced
P P P P

The quick sort algorithm is a very efficient sorting algorithm. On average, time complexity is $O(n \; log \; n)$ which is the same as a merge sort. However, in the best-case time complexity merge sort outperforms quick sort. Despite this, the quick sort is often favoured over merge sort.

Which of the following statements provides the best reason for choosing quick sort to sort your data?

- ○ Quick sort has better space complexity than merge sort.
- ○ Quick sort will out perform merge sort if an optimal pivot value is chosen.
- ○ Quick sort is easier to code than merge sort.
- ○ Quick sort performs better than merge sort in the worst-case time complexity.