

Binary search: time complexity

Challenge 1



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

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

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Merge sort: time complexity

Challenge 1



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

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

Quiz:

STEM SMART Computer Science Week
46

All teaching materials on this site are available under a [CC BY-NC-SA 4.0](#) license, except where otherwise stated.



Big O: rank complexities

Challenge 2



Put the following Big O expressions into the order of increasing complexity.

Available items

$O(\log n)$
$O(2^n)$
$O(n \log n)$
$O(1)$
$O(n^2)$
$O(n!)$
$O(n)$

Quiz:

STEM SMART Computer Science Week
46

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Intractable problem 1

Practice 2



What is an intractable problem?

- ☐ A problem for which the only known algorithms are impractical for large inputs.
- ☐ When it is not possible to tell how long the program that solves the problem will run for
- ☐ When a program that solves the problem has polynomial time complexity or less
- ☐ When it is not possible to write a program that can solve the problem

Quiz:

STEM SMART Computer Science Week
46

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Big O: determine expression 2

Challenge 2



The number of operations (steps) of an algorithm is given below, where n is the size of the input:

$$5n^2 + \log_2 n + 2^n + 18$$

How would the time complexity of this algorithm be expressed in **Big O notation**?

Type your answer in the format $\mathcal{O}(n)$, replacing n with the relevant term. If you want to use an exponent, use the \wedge symbol, for example, $\mathcal{O}(n^2)$.

Quiz:

STEM SMART Computer Science Week

46

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Insertion sort: time complexity

Challenge 1



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

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

Quiz:

STEM SMART Computer Science Week
46

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Binary search: trace

Challenge 2



The following list of items is stored in an array:

items							
index	0	1	2	3	4	5	6
value	2	4	5	8	12	15	18

The following algorithm has been coded and will be used to search for the number 12 as highlighted in the array.

Pseudocode

```
1 FUNCTION binary_search(items, search_item)
2   found = False
3   found_index = -1
4   first = 0
5   last = LEN(items) - 1
6
7   WHILE first <= last AND found == False
8     midpoint = (first + last) DIV 2
9     IF items[midpoint] == search_item THEN
10      found_index = midpoint
11      found = True
12     ELSEIF items[midpoint] < search_item THEN
13      first = midpoint + 1
14     ELSE
15      last = midpoint - 1
16     ENDIF
17   ENDWHILE
18
19   RETURN found_index
20 ENDFUNCTION
```

Complete the trace table for the algorithm. The first pass has already been filled for you.

first	last	midpoint	items [midpoint]	found_index	found
0	6	3	8	−1	False
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		False
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	4	True

Bubble sort: fix

Challenge 1



Kofi has written a program to sort an array of data into **descending** order. The algorithm is a version of bubble sort and is shown in pseudocode below.

Pseudocode

```
1 // Initialise variables
2 items = [28, 40, 21, 25, 30, 27, 25]
3 num_items = LEN(items)
4 temp = 0
5 pass_number = 1
6 swapped = False
7
8 // Continue while swaps have been made and there are more passes to evaluate
9 WHILE swapped == True AND pass_number <= num_items - 1
10     swapped = False
11     FOR index = 0 TO num_items - 2
12         // Check if items are out of order
13         IF items[index] < items[index + 1]
14             // Swap items
15             temp = items[index]
16             items[index] = items[index + 1]
17             items[index + 1] = temp
18             swapped = True
19         ENDIF
20     NEXT index
21     pass_number = pass_number + 1
22 ENDWHILE
```


Part A

There is a problem with the algorithm and it will not sort the data correctly. What is the problem?

- ☐ The variable **swapped** has been initialised incorrectly.
 - ☐ The **items** are being compared in the wrong order.
 - ☐ The outer loop is running insufficient times.
 - ☐ The inner loop is running too many times.
-
-

Part B

Kofi has corrected the error in the pseudocode.

What is the **total number of swaps** required to sort the array **items** into descending order?

Enter your answer as a **number**.

Quiz:

STEM SMART Computer Science Week

46

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Recursion: purpose of subroutine 2

Challenge 2



A recursive subroutine has been written in high-level pseudocode, as follows:

Pseudocode

```
1 PROCEDURE traverse(tree)
2   IF tree != Null THEN
3     traverse(tree.RightChild())
4     PRINT(tree.node.get_value())
5     traverse(tree.LeftChild())
6   ENDIF
7 ENDPROCEDURE
```

What **variant** of a standard algorithm does it implement?

Quiz:

[STEM SMART Computer Science Week 46](#)

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Features of recursion

Challenge 2



Recursion is a technique often used in programming to solve complex problems by breaking them down into smaller, similar subproblems.

State two features of recursion in programming.

[2 marks]

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

