B2.4

Programming algorithms

CHAPTER ASSIGNMENT

Name _____ Class _____ No. _____ Date _____

Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Computer Science for the IB Diploma*, 1st edition (2025), Paul Baumgarten, Ioana Ganea, Carl Turland (Hachette Learning), for classroom use.

Review Questions — Big O Notation (p. 361)

1. Explain what Big O notation is and why it is used in Computer Science.

2. Define the terms "time complexity" and "space complexity".

3. Outline the difference between $O(n)$ and $O(n^2)$. Identify two algorithms that present these complexities.

4. Explain why you think it is important to consider the worst-case space complexity of an algorithm.

5. Sketch a graph to compare the time complexity of $O(n)$ and $O(\log n)$. What differences do you notice?

6. Consider the following statement: "An algorithm with a Big O value of $O(n)$ will always be slower than one with a value of $O(\log n)$". Evaluate this statement to true or false and explain your reasons.

7. Outline the two types of complexity analysis.

8. Identify which is generally faster for a very large n: $O(\log n)$ or $O(2^n)$.

Review Questions — Recursion (HL) (p. 371)

1. Define the term "recursion".

2. Compare recursive and iterative algorithms.

3. Explain the need for a base case in a recursive program.

4. Identify one consequence of not using a base case in a recursive function.

5. Identify some advantages and disadvantages of using recursion.

6. Explain what steps you would take to debug a recursive function that does not work as expected.

Review Questions — Sorting Algorithms (HL) (p. 375)

1. Complete a trace table for a bubble sort algorithm to sort the numbers: `16, 13, 4, 6, 22, 1, 9, 5`.

2. Complete a trace table for a selection sort algorithm to sort the numbers: `16, 13, 4, 6, 22, 1, 9, 5`.

3. Complete a trace table for a quicksort algorithm to sort the numbers: `16, 13, 4, 6, 22, 1, 9, 5`.

4. Create a table outlining the Big O value for both time and space complexities for all the searching and sorting algorithms you have studied.

Algorithm	Time Complexity	Space Complexity
Linear search		
Binary search		
Bubble sort		
Selection sort		
Quicksort		

5. Outline the principles of working of a selection sort and a quicksort algorithm.

Programming Exercise — Binary Tree Traversal (HL) (p. 376)

1. Implement the post-order and pre-order traversals of a binary tree using recursion.

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/ib-computer-science/b2-4-algorithms/