

Topic 19: Computational Thinking and Problem-Solving

EAL keyword sheet - Cambridge International AS & A Level Computer Science 9618. Write the word in your own language in the last column.

Keyword	Meaning	Example sentence	In my language
linear search	Checks each item in order.	<i>The linear search found the item at position 4.</i>	
binary search	Halves a sorted list each time.	<i>Binary search is fast on big lists.</i>	
insertion sort	Builds a sorted list by inserting each item in its place.	<i>Insertion sort is good for small lists.</i>	
bubble sort	Swaps neighbouring items until the list is sorted.	<i>Bubble sort is simple but slow.</i>	
abstract data type (ADT)	A data structure defined by its operations.	<i>A queue is an ADT.</i>	
stack	Last in, first out. Operations: push, pop.	<i>A stack is used for undo.</i>	
queue	First in, first out. Operations: enqueue, dequeue.	<i>A print queue is a queue.</i>	
circular queue	A queue where the end links back to the start of the array.	<i>A circular queue reuses empty space.</i>	
linked list	Nodes joined by pointers.	<i>A linked list can insert items quickly.</i>	
head pointer / null pointer	Points to the first node / points to nothing.	<i>The last node has a null pointer.</i>	
binary tree	A tree in which each node has at most two children.	<i>A binary tree can store ordered data.</i>	
root	The top node of a tree.	<i>Start the search at the root.</i>	
leaf	A node with no children.	<i>A leaf node has no child.</i>	
traversal	Visiting every node in a tree: pre-order, in-order, post-order.	<i>In-order traversal gives sorted output.</i>	
dictionary	A structure of key-value pairs.	<i>A dictionary finds a value by its key.</i>	
graph	Nodes and edges.	<i>A graph can represent a network.</i>	
adjacency matrix / list	Ways of storing a graph.	<i>An adjacency matrix is a 2D array.</i>	
Big O notation	A way to describe how the time or space needed grows with the input size.	<i>Linear search is $O(n)$.</i>	

Keyword	Meaning	Example sentence	In my language
time complexity	How the run time grows as the input grows.	<i>Binary search has $O(\log n)$ time complexity.</i>	
space complexity	How the memory needed grows as the input grows.	<i>Recursion can have high space complexity.</i>	
recursion	A subroutine that calls itself.	<i>A recursive function calls itself.</i>	
base case	The condition that stops recursion.	<i>Without a base case, recursion never ends.</i>	
general case	The part of a recursive function that calls itself.	<i>The general case reduces the problem.</i>	
stack overflow	An error when too many recursive calls fill the stack.	<i>Infinite recursion causes a stack overflow.</i>	
iteration	Repeating using loops.	<i>Iteration usually uses less memory than recursion.</i>	
unwinding	Returning values back up the chain of recursive calls.	<i>The answer appears during unwinding.</i>	