Definitions and Concepts for OCR Computer Science A-level

Component 2.3: Algorithms

2.3.1 Algorithms

A* Algorithm: A generalisation of Dijkstra's algorithm that approximately finds the shortest path between two nodes by keeping track of the distance travelled from the source to the current node (cost function) and the approximate distance from the next node to the destination (heuristic function).

Big O Notation: A way of classifying algorithms based on their complexity by setting an upper bound for the worst case space or time complexity as a mathematical function of input size.

Binary Search: A O(log(n)) algorithm to search a sorted list for a particular item by repeatedly halving the sublist which could contain the item.

Breadth-First Traversal : A method of traversing a graph by using a queue to visit all the neighbours of the current node before doing the same to each of the neighbours until the entire graph has been explored.

Bubble Sort: A $O(n^2)$ sorting algorithm that iterates through a list, comparing each element to its successor and swapping elements if the successor is greater than the current element. This is repeated until no more swaps can be made.

Depth-First Traversal: A method of traversing a graph by using a stack to travel as far along one route as possible and then backtracking and doing the same for the remaining routes until the entire graph has been explored.

Dijkstra's Shortest Path Algorithm: An algorithm to find the shortest path between two nodes on a graph by using a priority queue to keep track of the shortest distance (cost) to each node from the starting node until the destination node is found.

Insertion Sort: A $O(n^2)$ sorting algorithm that divides a list into a sorted and unsorted section. Elements from the unsorted section are compared with and placed in the right position in the sorted section one by one until the sorted section spans the entire list.

Linear Search: A O(n) algorithm to search a list for a particular item by iterating through the list and checking each element until the required item is located, or the end of the list is reached.

Merge Sort: A O(n log(n)) divide-and-conquer sorting algorithm that recursively halves the list into sublists until all sublists are of length 1. The sublists are then merged back together

in such a way that they are always sorted, until the full single list is obtained.

O(1) (Constant time/space): An algorithm that always takes a constant amount of time or memory space to execute, regardless of the input size.

O(2^n) (Exponential time/space): An algorithm whose execution time or required memory space grows exponentially with input size (higher complexity than polynomial).

O(log(n)) (Logarithmic time/space): An algorithm whose execution time or required memory space grows logarithmically with input size (lower complexity than polynomial).

O(n) (Linear time/space): An algorithm whose execution time or required memory space is directly proportional to the size of its input.

O(n^k) (Polynomial time/space): An algorithm whose execution time or required memory space is proportional to the input size raised to the power of a constant (k). Eg. O(n^2), O(n^3) etc.

Quick Sort: A O(n log(n)) sorting algorithm similar to Merge Sort, but relies on choosing a pivot element whenever the list is split. Elements greater than the pivot go into one sub list, while the rest go into the other. All sublists of length 1 will already be in a sorted order.

Space Complexity: A measure of the amount of memory space needed by an algorithm to solve a particular problem of a given input size.

Time Complexity: A measure of the amount of time needed by an algorithm to solve a particular problem of a given input size.