AQA Computer Science A-Level

4.3.4 Searching algorithms

Concise Notes

Specification:

4.3.4.1 Linear search:

Know and be able to trace and analyse the complexity of the linear search algorithm. Time complexity is O(n).

4.3.4.2 Binary search

Know and be able to trace and analyse the time complexity of the binary search algorithm. Time complexity is O(log n).

4.3.4.3 Binary tree search

Be able to trace and analyse the time complexity of the binary tree search algorithm. Time complexity is O(log n).

Searching Algorithms

- A searching algorithm is used to find a specified data item within a set of data

Linear Search

- Can be conducted on any list, even if the data isn't in order
- Works by inspecting every item in a list one by one until the desired item is found
- Very simple to program
- Comparatively high time complexity of O(N)
- If the target does not exist in the list, the algorithm will still check every single item in the list

Binary Search

- Can only be used on ordered lists
- Works by looking at the midpoint of a list and determining if the target is higher or lower than the midpoint
- More efficient than the linear search algorithm
- Midpoint calculated by adding the first and last positions of the data, and dividing by two
- The list of data is halved with each iteration
- Can be implemented both iteratively and recursively
- Good time complexity of O(logN)

Binary Tree Search

- The same as a binary search, but conducted on a binary tree
- A binary tree is a rooted, ordered tree in which each node has no more than 2 children
- Can be implemented both iteratively and recursively
- Same time complexity as binary search, O(logN)