



S1_01 Data Structures And Algorithms : Subject Content

Course Menu

Zoom Size: 100% ✔ < Previous Page: 1 / 10 Next >

Reading Material



ALGORITHM ANALYSIS AND DESIGN **CONCEPTS**

Algorithm Analysis

Table of Contents

- Algorithm Analysis
 - 3.1 Analysis of Searching Algorithms Linear Search
 - 3.2 Analysis of Searching Algorithms Binary Search
 - 3.3 Analysis of Sorting Algorithms Bubble Sort
 - 3.4 Analysis of Sorting Algorithms Selection Sort
 - 3.5 Analysis of Sorting Algorithms Insertion Sort
 - 3.6 Analysis of Sorting Algorithms Heap Sort
 - 3.7 Analysis of Sorting Algorithms Merge Sort

3. Algorithm Analysis

In this section, we will look at some selected algorithms and understand their performance characteristics.

3.1 Analysis of Searching algorithm - Linear Search

The linear search algorithm is applied to an unordered set of elements.

Let us conduct a worst-case analysis of this algorithm - in this case, the search element is either at the last position or is not present in the array. The total number of searches required in this scenario is n, where n is the total number of elements. If initialization and assignment operations take a constant time, k, the worst-case complexity would be a summation of this number k and the time taken for n searches.

Hence the worst-case complexity is T(n) which is equal to $O\left(n+k\right)$ = $O\left(n\right)$

3.2 Analysis of Searching algorithm - Binary Search

A binary search algorithm is applied only on a sorted list of items. Low and high are two indexes used, which are initialized to one and N, respectively, and S stores the item to be searched.

Views: 15686

230 📭 0

< qsort() Library Function Demo (/?q=MULNCourseBook/listDetailedCBLearningContents/178099/cidb/full/view/153784/431550/431535/39731)