Searching & Sorting Algorithms

Tags	Algorithm, Elementary-Sorting_Algorithms, Sorting
Companies	
Created time	@November 3, 2022 1:48 PM
Last edited time	@November 5, 2022 6:07 AM
Status	Not started
URL	
URL 1	https://medium.com/interviewnoodle/search-sort-a-algorithm-ride-7919b1923585

SEARCHING:-

Searching

Not even a single day pass, when we do not have to search for something in our day-to-day life, car keys, books, pen, mobile charger, and whatnot. The same is the life of a computer, there is so much data stored in it, that whenever a user asks for some data, the computer has to search its memory to look for the data and make it available to the user. And the computer has its own techniques to search through its memory fast.

Well, to search for an element in a given array, there are two popular algorithms available:

1) Linear Search
2) Binary Search

LINEAR SEARCH:-

Linear Search

Linear search is a very basic and simple search algorithm. In Linear search, we search for an element or value in a given array by traversing the array from the starting, till the desired element or value is found. It compares the element to be searched with all the elements present in the array and when the element is

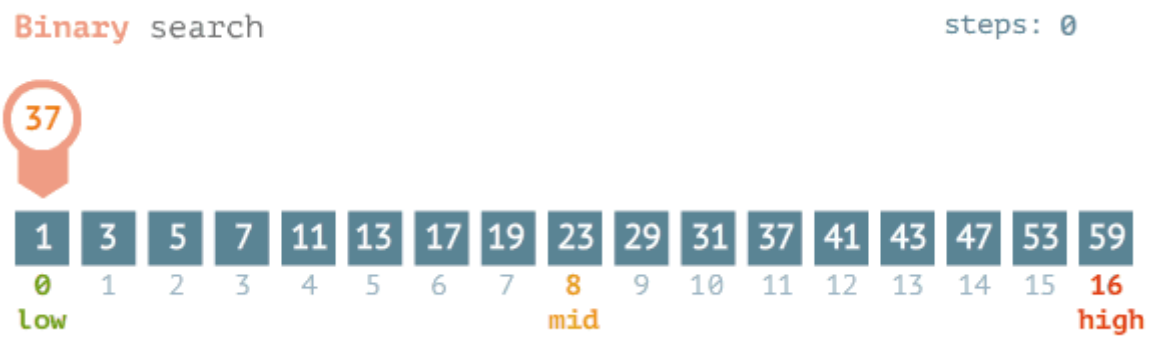
matched successfully, it returns the index of the element in the array, else it returns -1. Linear Search is applied on unsorted or unordered lists when there are fewer elements in a list.

Feature's of Linear Search:-

1) It is used for an unsorted and unordered small list of elements.
2) It has a time complexity of O(n), which means the time is linearly dependent on the number of elements, which is not bad, but not that good too.
3) It has a very simple implementation.

BINARY SEARCH:-

Binary Search



Binary Search is applied on the sorted array or list of large size. Its time complexity of O(log n) makes it very fast as compared to other sorting algorithms. The only limitation is that the array or list of elements must be sorted for the binary search algorithm to work on it.

Following are the steps of implementation that we will be following:

1) Start with the middle element:
2) If the target value is equal to the middle element of the array, then return the index of the middle element.
3) If not, then compare the middle element with the target value,
4) If the target value is greater than the number in the middle index, then pick the elements to the right of the middle index, and start with Step 1.
5) If the target value is less than the number in the middle index, then pick the elements to the left of the middle index, and start with Step 1.
6) When a match is found, return the index of the element matched.
7) If no match is found, then return -1.

Feature's of Binary Search:-

1) It is used for only sorted arrays and the size of the array doesn't matter.

2) It has a time complexity of ***O(log N)***, which means the time is not linearly dependent on the number of elements.

INDEXED SEQUENTIAL SEARCH:-

INDEXED SEQUENTIAL SEARCH

Index search is a special search. This search method is used to search a record in a file. Searching a record refers to searching for the location loc in the memory where the file is stored. Indexed search searches the record with a given key value relative to a primary key field. This search method is accomplished by the use of pointers. The index helps to locate a particular record with less time. Indexed sequential files use the principle of index creation.

Features of Indexed Sequential Search:-

1. In Indexed Sequential Search a sorted index is set aside in addition to the array. 2) Each element in the index points to a block of elements in the array or another expanded index. 3) The index is searched 1st then the array and guides the search in the array.

SORTING:-

Sorting: By Netflix

Now let’s talk about some of the popular sorting techniques. There are so many things in our real life that we need to search for, like a particular record in the database, roll numbers in the merit list, a particular telephone number in the telephone directory, a particular page in a book, etc. All this would have been a mess if the data was kept unordered and unsorted, but fortunately, the concept of sorting came into existence, making it easier for everyone to arrange data in order, hence making it easier to search. Sorting arranges data in a sequence which makes searching easier.
so what is the real meaning of sorting:- ***Sorting*** is nothing but arranging the data in ascending or descending order. The term sorting came into the picture, as humans realized the importance of searching quickly.

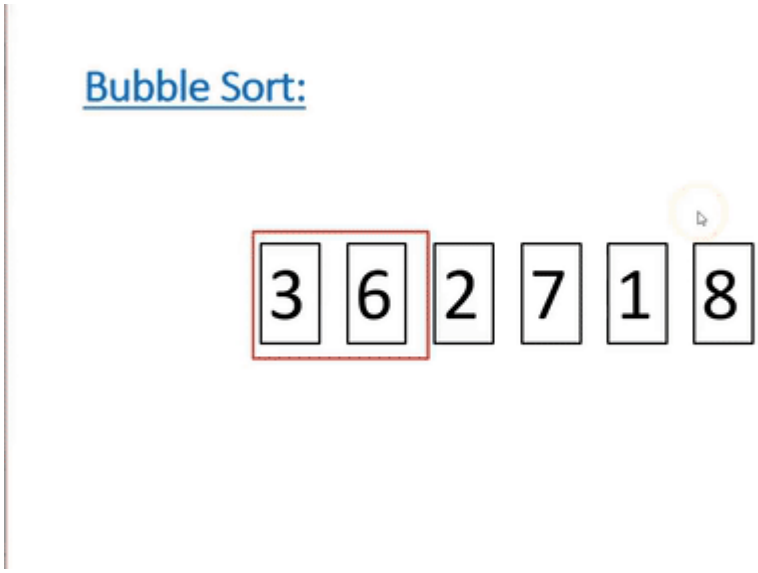
Some of the popular sorting techniques are as follows:-

1) Bubble Sort
2) Insertion Sort
3) Selection Sort
4) Merge Sort
5) Heap Sort

Let’s discuss this sorting one by one in detail:-

BUBBLE SORT:-

Bubble sort



Bubble Sort is a simple algorithm that is used to sort a given set of n elements provided in form of an array with n number of elements.

Bubble Sort compares all the elements one by one and sorts them based on their values. If the given array has to be sorted in ascending order, then bubble sort will start by comparing the first element of the array with the second element, if the first element is greater than the second element, it will swap both the elements, and then move on to compare the second and the third element, and so on.

It is known as bubble sort, because with every complete iteration the largest element in the given array, bubbles up towards the last place or the highest index, just like a water bubble rises up to the water surface.

Sorting takes place by stepping through all the elements one-by-one and comparing them with the adjacent element and swapping them if required.

Feature's of Bubble sort:-

1. The time complexity of Bubble Sort is O(n2).
2. The space complexity for Bubble Sort is O(1), because only a single additional memory space is required i.e. for the temp variable.
3. Also, the best case time complexity will be O(n), which is when the list is already sorted.

INSERTION SORT:-

INSERTION SORT

Insertion sort works as sorting a deck of cards. It starts from index 1(not 0), and each index starting from index 1 is like a new card, that you have to place at the right position in the sorted subarray on the left.

It is efficient for smaller data sets but very inefficient for larger lists. Insertion Sort is adaptive, which means it reduces its total number of steps if a partially sorted array is provided as input, making it efficient. It is better than Selection Sort and Bubble Sort algorithms. Its space complexity is less.

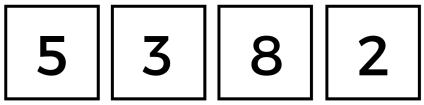
Like bubble sort, insertion sort also requires a single additional memory space. It is a stable sorting technique, as it does not change the relative order of elements that are equal.

Feature's of Insertion sort :-

1) Worst CaseTime Complexity O(n2).
2) Space Complexity: O(1)

SELECTION SORT:-

SELECTION SORT



Selection sort is conceptually the simplest sorting algorithm. This algorithm will first find the smallest element in the array and swap it with the element in the first position, then it will find the second smallest element and swap it with the element in the second position, and it will keep on doing this until the entire array is sorted. It is called selection sort because it repeatedly selects the next smallest element and swaps it into the right place.

Selection Sort requires two nested for loops to complete themselves.

Feature's of Selection Sort:-

1. Worst-case time complexity O(n2).
2. Average time Complexity θ(n2).
3. Space Complexity: O(1).

MERGE SORT:-

Merge Sort follows the rule of Divide and Conquer to sort a given set of numbers/elements, recursively, hence consuming less time. Merge sort runs in O(n*log n) time in all the cases. Before jumping on to, how merge sort works and its implementation, first let's understand what is the rule of Divide and Conquer?

Divide and Conquer:-

Divide & Conquer

6 5 3 1 8 7 2 4

If we can break a single big problem into smaller subproblems, solve the smaller sub-problems and combine their solutions to find the solution for the original big problem, it becomes easier to solve the whole problem. In Merge Sort, the given unsorted array with n elements is divided into n sub-arrays, each having one element because a single element is always sorted in itself. Then, it repeatedly merges these sub-arrays, to produce new sorted sub-arrays, and in the end, one complete sorted array is produced.

The concept of Divide and Conquer involves three steps:

1) Divide the problem into multiple small problems.

2) Conquer the subproblems by solving them. The idea is to break down the problem into atomic subproblems, where they are actually solved.

3) Combine the solutions of the subproblems to find the solution to the actual problem.

Feature's of Merge Sort:-

1. Worst Case Time Complexity O(n*log n).
2. Average Time Complexity θ(n*log n).
3. Space Complexity: O(n).

HEAPSORT:-

HEAPSORT

6 5 3 1 8 7 2 4

Heap Sort is one of the best sorting methods being in place and with no quadratic worst-case running time. Heap sort involves building a Heap data structure from the given array and then utilizing the Heap to sort the array. Heap is a special tree-based data structure, that satisfies the following special heap properties:

Shape Property: Heap data structure is always a Complete Binary Tree, which means all levels of the tree are fully filled.***Heap Property:*** All nodes are either greater than or equal to or less than or equal to each of its children. If the parent nodes are greater than their child nodes, the heap is called a MaxHeap, and if the parent nodes are smaller than their child nodes, the heap is called Min-Heap.

Features of Heapsort:-

1. Worst Case: O(n log n).
2. Average Case: O(n log n).