Binary Search

Difficulty Level : Easy

Last Updated : 03 Aug, 2022

Problem: Given a sorted array arr[] of n elements, write a function to search a given element x in arr[] and return the index of x in the array.

Consider array is 0 base index.

Examples:

Input: arr[] = {10, 20, 30, 50, 60, 80, 110, 130, 140, 170}, x = 110

Output: 6

Explanation: Element x is present at index 6.

Input: arr[] = {10, 20, 30, 40, 60, 110, 120, 130, 170}, x = 175

Output: -1

Explanation: Element x is not present in arr[].

Linear Search Approach: A simple approach is to do a linear search. The time complexity of the Linear search is O(n). Another approach to perform the same task is using Binary Search.

Binary Search Approach:

Binary Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n).

Binary Search Algorithm: The basic steps to perform Binary Search are:

Begin with the mid element of the whole array as a search key.

If the value of the search key is equal to the item then return an index of the search key.

Or if the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half.

Otherwise, narrow it to the upper half.

Repeatedly check from the second point until the value is found or the interval is empty.