## Binary Search

1. In computer science, a **binary search** or half-interval **search** algorithm finds the position of a specified input value (the **search** "key") within an array sorted by key value. For **binary search**, the array should be arranged in ascending or descending order
2. It is efficient
3. We have to sorted array(requirement)
4. Sequential Search on an Ordered File

**Sequential Searches**

In computer science, **linear search** or sequential **search** is a method for finding a particular value in a list, that consists of checking every one of its elements, one at a time and in sequence, until the desired one is found.

1. Compare each element in sequence
2. No precondition that we have to sorted array
3. Worse case is we have to search the whole list
4. Liner search is easy to program
5. Sequential Search on an Unordered File

Sorting involves arranging data in ascending or descending order, according to a certain collating sequence (or sorting sequence). The sorting algorithm includes:

* Insertion Sort: See "[Insertion Sort](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp1_Basics.html" \l "InsertionSort)".
* Selection Sort: See "[Selection Sort](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp1_Basics.html#SelectionSort)".
* Bubble Sort: See "[Bubble Sort](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp1_Basics.html#BubbleSort)"
* Merge Sort (Recursive Top-Down or Interactive Bottom-up)
* Quick Sort (Recursive)
* Bucket Sort
* Heap Sort
* Binary Tree Sort

Searching

Computer systems are often used to store large amounts of data from which individual records must be retrieved according to some search criterion. Thus the efficient storage of data to facilitate fast searching is an important issue. In this section, we shall investigate the performance of some searching algorithms and the data structures which they use