Differences between Linear search and Binary search

**Let's look at the differences in a tabular form.**

|  |  |  |
| --- | --- | --- |
| **Basis of comparison** | **Linear search** | **Binary search** |
| **Definition** | The linear search starts searching from the first element and compares each element with a searched element till the element is not found. | It finds the position of the searched element by finding the middle element of the array. |
| **Sorted data** | In a linear search, the elements don't need to be arranged in sorted order. | The pre-condition for the binary search is that the elements must be arranged in a sorted order. |
| **Implementation** | The linear search can be implemented on any linear data structure such as an array, linked list, etc. | The implementation of binary search is limited as it can be implemented only on those data structures that have two-way traversal. |
| **Approach** | It is based on the sequential approach. | It is based on the divide and conquer approach. |
| **Size** | It is preferrable for the small-sized data sets. | It is preferrable for the large-size data sets. |
| **Efficiency** | It is less efficient in the case of large-size data sets. | It is more efficient in the case of large-size data sets. |
| **Worst-case scenario** | In a linear search, the worst- case scenario for finding the element is O(n). | In a binary search, the worst-case scenario for finding the element is O(log2n). |
| **Best-case scenario** | In a linear search, the best-case scenario for finding the first element in the list is O(1). | In a binary search, the best-case scenario for finding the first element in the list is O(1). |
| **Dimensional array** | It can be implemented on both a single and multidimensional array. | It can be implemented only on a multidimensional array. |