**Big O notation**

Big O notation is a mathematical way of describing the space and time complexity of an algorithm used in the code in terms of the size of the input, which is generally denoted as **n**. It tells us how fast an algorithm works with an increase in input size and the efficiency of the algorithm.

**Linear Search**

This is a searching technique that checks each and every item of the array to see if the item is present in the array or not. This is a slow process, but the item will surely be found (if it is present in the array). This works on both sorted and unsorted array.

* Best case (First element target): O(1)
* Average case (Target in the middle): O(n)
* Worst case (Target at end or no target not found): O(n)

**Binary Search**

This is a searching technique that starts from the middle of a sorted array to check the presence of the target. This is a fast process since it divides the search space in half. This works on a sorted array only.

* Best case (Middle element is the target): O(1)
* Average case (the middle value being divided in half to reach the target): O(log n)
* Worst case (Full division of the middle value to reach the target): O(log n)

**Conclusion**

Linear search should be used for small or unsorted array whereas Binary Search should be used for large, sorted data for best performance.