**Linear search**

* Linear search is an algorithm to search an element from a list sequentially, which means it starts from the starting point and directly search each element until the required element is reached.
* This method of Linear search takes more time that is where the Binary search comes in to action.
* Efficiency: The time complexity of linear search is O(n), where n is the number of elements in the list. This makes it less efficient for large lists compared to binary search.

**Binary search in Java**

* Binary search is an algorithm used for finding a specific item from a sorted list.
* It finds the required item by cut shorting the range where the search is going to happen.
* It works only with a sorted list.

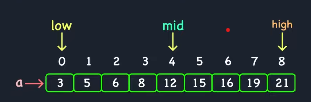
**Key Concepts of Binary Search**

* **Sorted List Requirement:** Binary search only works on lists that are sorted. This is because the algorithm relies on comparing the target value with the middle element to decide which half of the list to search next.
* **Divide and Conquer:** Binary search reduces the search space by half with each step, which makes it much faster than a linear search, especially for large datasets.
* **Efficiency**: For sorting (If array not sorted)= O(n log n) and for search = O(log n) .

**How Binary search works:**

* Consider 3 reference points in the sorted array say, **Low**, **high** and **mid.**
* The value we are searching is the **Key.**

Where mid = (low high)/2



* Step 1 :Check whether the element you are searching is the middle element or not.

If (key==a [mid]

{

Return mid

}

* Step 2 : If the element you search is less than the middle element.

else if (key < a[ mid])

{

In case the key value is less than the middle value adjust low and high in this method

High =mid-1

Low = low

}

* Step 3 : If the element you search is greater than the middle element.

else

{

In case the key value is greater than the middle value adjust low and high in this method

low = mid +1

High = high

}

