Searching Algorithms

By Dr Shantanu Pathak

Types of Searching

- Linear Search
- Binary Search

- Hashing using Hash Table

Linear Search

- Search the element by traversing elements one by one
- Beginning to end all elements are checked one by one

- Slow if very high number of elements are stored

- Takes O(n) time
    - Why O(n) ?
- **Example :**
- Searching element in Array / linked list
- [10, 45, 76, 43, 12, 32] search 12 in these elements
- Start from 10, keep on checking one element at a time in a loop

Binary Search

- Fast searching
- Every step reduces the number of elements by half
- Assumes input elements are sorted
- At every step check with the middle element of currently selected part of elements
- If matched then return the matched index
- Else , select left side half or right side half part to search

- Takes O(log n) time

Binary Search Algorithm

- Input : Sorted array/list of elements
- Initialization: lowIndex =0, highIndex=arr.length-1, midIndex=-1
- Step 1. midIndex = (lowIndex + highIndex) /2
- Step 2. if arr[midIndex] == searchElement then return midIndex
- Step3. else-if searchElement < arr[midIndex]
  - //Search ONLY left part of the current array , So
  - highInedx = midIndex -1
  - Else ::
  - // Search ONLY right part of the current array, So
  - lowIndex = midIndex + 1
- Step 4. if (lowIndex <= highIndex) --> go to **step 1**