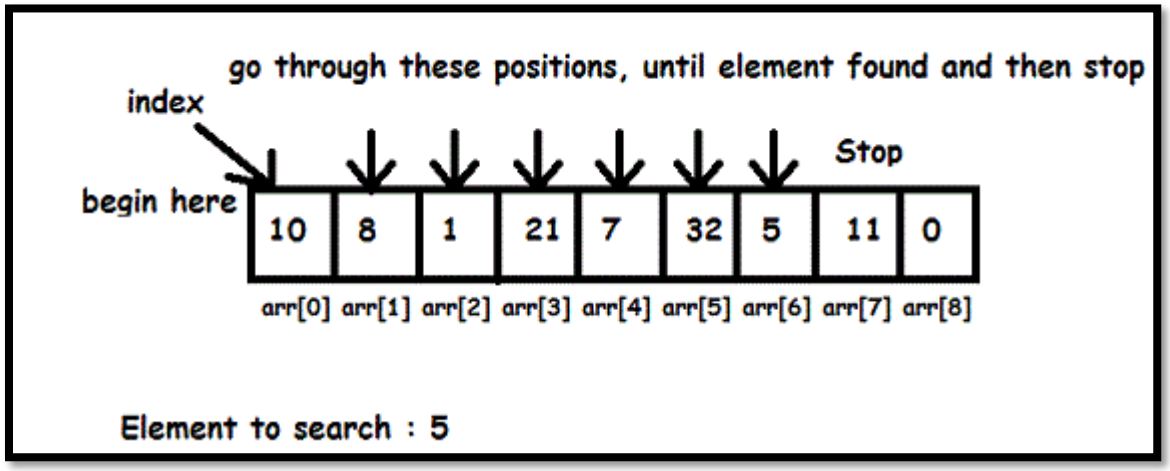
Lab#03-Implementation of Searching Algo(Array)

Objective

Implement searching algorithm (Array)

Linear Search

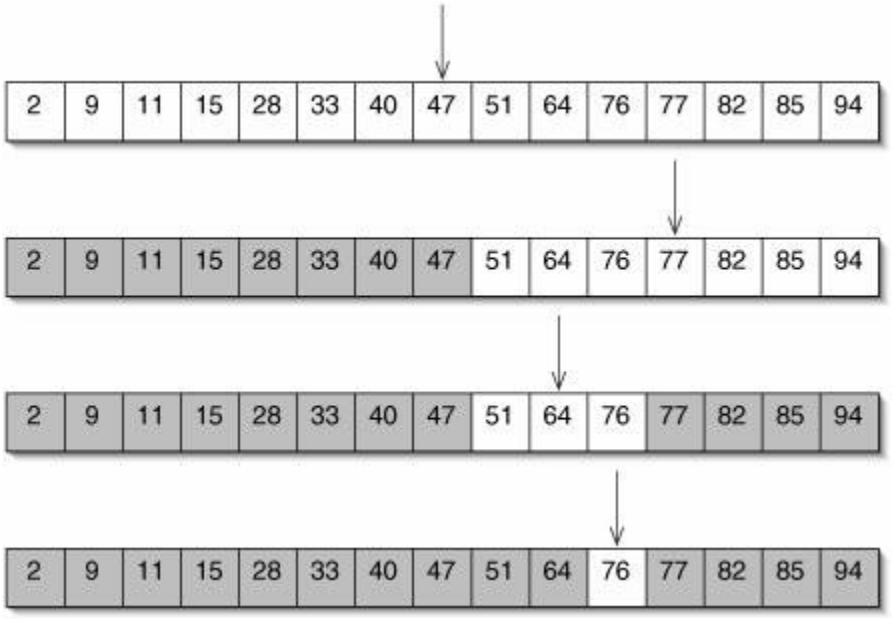


Linear Search: Steps on how it works:

1. Here is simple approach is to do Linear Search:
2. Start from the leftmost element of array and one by one compare the element we are searching for with each element of the array.
3. If there is a match between the element we are searching for and an element of the array, return the index.
4. If there is no match between the element we are searching for and an element of the array, return -1.

Binary Search

Happy coding ☺



Binary Search: Steps on how it works:

1. To find the index of element e with a certain value:
2. Start with an array sorted in descending order.
3. In each step: Pick the middle element of the array m and compare it to e. If element values are equal, then return index of m. If e is greater than m, then e must be in left subarray. If m is greater than e, then e must be in the right subarray.
4. Repeat those steps on new subarray.

Comparison between Binary Search and Linear Search:

- Binary Search requires the input data to be sorted; Linear Search doesn't
- Binary Search requires an ordering comparison; Linear Search only requires equality comparisons
- Binary Search has complexity O(log n); Linear search has complexity O(n) as discussed earlier
- Binary Search requires random access to the data; Linear Search only requires sequential access (this can be very important — it means a Linear Search can stream data of arbitrary size)

How to calculate the execution time(milliseconds) of a method in java:

```
  long before1=System.nanoTime ();
  Searching_lab.greeting1();
  long after1 = System.nanoTime();
  System.out.println("Execution time of greeting1 method is : "+(after1-
before1));
```

Happy coding ☺

Exercises

Task#01

Implement linear search on 1D and 2D array.
Note: create methods for both 1D and 2D linear search

//LinearSearch1D(int[] A, int key)

//LinearSearch2D(int[][] A, int key)

//call methods and show the search result

Task#02

Implement binary search on 1D and 2D array.
Note: create methods for both 1D and 2D binary search

//BinarySearch1D(int[] A, int key, int first, int end)

//BinarySearch2D(int[][] A, int key, int first, int end)

//call methods and show the search result

Task#03

Display the execution time of searching algo (linear and binary both) and examine which one is the faster and explain why?

```
 long before1=System.currentTimeMillis();
     //call method here
 long after1=System.currentTimeMillis();
//than fine difference between after and before variable – it will give you the
execution time of method
```

Sample Output:

Time took by Linear Search : ____ (time in milliseconds or nanoseconds)

Time took by Binary Search : ____ (time in milliseconds or nanoseconds)

Happy coding ☺