* Algorithm

Linear Search

```
linear search (arr[], ~~size~~, key)
    for (i=0 to i=n-1, i++)
        if (arr[i] == mid)
            return i;      // index of key
    return -1;
```

// Aim: find index of key by linear search.

// input: array (size & key)

// output: index of number is found
or
return -1;

Binary Search.

// Aim: Find index of key by binary Search

// input: sorted array of integer, size and key.

// output: index of number is found
or
return -1;

```
start = 0, end = size - 1, mid = $\frac{start + 1}{2}$

        while ( start <= end)
       { if (arr [mid] == key)
              return mid
            if (arr [mid] < key)
                start = mid + 1;
            else
                  end = mid - 1;
         }
            return -1;
         }.
```

Binary Search Recurrsive

```
binary s ( arr [], start, end, key)
    { mid ← $\frac{start + end}{2}$

          if (start > end)
              return -1;
            if (arr [mid] == key)
               return mid;
             if (arr [mid] < key)
       binary s ( arr [], mid + 1, end, key)
       else
```

```
        {
          binary s (arr [], start, mid-1, key)
        }
      }
```

Test Cases (Binary Search)

1) input arr = { 1, 2, 3, 4, 5, 6, 7 }
Key = 3
expected output = "nuber found at index 2"

actual output : "numbes found at index : 2"

2) input arr = { 10, 20, 30, 40, 50 }
Key = 40.
expected out-put = "number found at index 3"
actual out-put = " number found at index 3"

3) input arr = { 5, 10, 12, 13 17, 30 }
Key = 5
expected output = "number found at index 0"
Actual output = " number found at index 0".

4) input arr = { 1, 3, 5 }

Key = 8

expected output = " number not found".

actual output = " number not found"

5) input arr = { 4, 7, 11, 16 }

Key = 5

expected output = " number not found"

actual output = " numbes not found".

Test cases (linear search)

1) input arr = { 1, 8, 5, 3, 10, 11 }

Key = 3

expected output = {" number found at index 3"

~~ex~~actual output = " number found at index 3".

2) index arr = { 8, 18, 28, 38, 48 }

Key = 28

expected output = " number found at index 2"

actual output = " number found at index 2"

__/__/____

3) input arr = { 5, 4, 8, 7 }
key = 17
expected output = " ~~key~~ number not found".
actual output - " number not found".

4) input arr = { 7, 17, 27, 13, 23, 3 }
key = 8
expected output - " number not found"
actual output = " number not found".

5) input arr = { }
key = 1
expected arr = " number not found"
actual arr = " number not found".

* Time Complexity.

- Linear complexity

Best case = $\Omega(1)$
Worst Case = $O(n)$

- Binary search complexity.

Best case = $\Omega(1)$
Worst Case = $O(\log n)$.