Search Algorithm

■ Linear Search

- Problem:

  - Input: A = <$a_1, a_2, \cdots, a_n$> , v
  - Output: index i such that $v = A[i]$

    NIL if v does not appear in $A[i]$

- Idea:

  - 순차적으로 i 를 1부터 n까지 증가시키며 $A[i]$ 탐색
  - incremental approach

- Algorithm

```
LINEAR-SEARCH($A, v$)
for $i \leftarrow 1$ to $n$
   do if ($A[i] = v$)
         return $i$;
return NIL;
```

- Analysis

  running time T(n) = $\Theta(n)$ (worst case)

■ Binary Search

- Problem:

  - Input: A = <$a_1, a_2, \cdots, a_n$> , v (단, $a_1 \le a_2 \le \cdots \le a_n$)

  - Output: index i such that $v = A[i]$

    NIL if v does not appear in $A[i]$

- Idea:

  - 중간 값을 탐색하여, v 보다 작으면 오른쪽으로, 크면 왼쪽으로 진행

  - divide-and-conquer approach

- Algorithm

```
BINARY-SEARCH($A, v$, left, right)
if (left <= right)
do
    middle = (left + right)/2;
    if (A[middle] < v)
        BINARY-SEARCH(A, v, middle+1, right)
    else if(A[middle] > v)
        BINARY-SEARCH(A, v, left, middle-1)
    else
        return middle
return NIL;
```

- Analysis

  running time T(n) = $\Theta(\log n)$ (worst case)