

# Recursion :

The process in which a function calls itself is called recursion & the function is called recursive function.

It is based on the mathematical concept called Principle of Mathematical Induction.

Prove :  $n=0$  or  $n=1$

Assume :  $n=k$  is true

Prove :  $n=k+1$  is true

Induction step : main problem ( $n$ )

Induction Hypothesis : sub problem ( $n-1$ )

We have,

base case

recursive call

small calculation

Recursive calls are stored in stacks & recursion depth might be exceeded.

## Data Structure

Linear

Dynamic

Non-Linear

Array  
Static

Linked  
List

Stacks

Queues

Tree

Graphs

Tables

Sets



# Must know search & sorting algorithms:

## 1. Binary Search:

- Array should be sorted
- Start by comparing the element ~~by~~ with the middle element of array
- if equals, return true
- if less than, I part of array
- if greater than, II part of array
- Repeat till found, else false.

code:

```
def binarysearch(a, x, si, ei):
```

```
    if si > ei:
```

```
        return -1
```

```
    mid = (si + ei) // 2
```

```
    if a[mid] == x:
```

```
        return mid
```

```
    elif a[mid] > x:
```

```
        return binarysearch(a, x, si, mid-1)
```

```
    else:
```

```
        return binarysearch(a, x, mid+1, ei)
```



## 2. Linear Search:

→ we search in each element of array linearly.

def linearSearch (a, si, ei, x):

if  $si > ei$ :

return -1

if  $a[si] == x$ :

$\Rightarrow O(n)$

return si

$\Rightarrow T(n-1) + K$

if  $a[ei] == x$ :

$\Rightarrow O(n)$

return ei

$\Rightarrow R(1)$

return linearSearch (a, si+1, ei-1, x)

## Important sorting algorithms:

1. Selection sort

2. Insertion sort

3. Quick sort

4. Merge sort

5. Bubble sort

6. Radix sort

7. Heap sort

8. Count sort

9. Shell sort

10. Bucket sort

11. Tower of Hanoi