(•) Calling a function inside itself is called as **recursion**. Such a function is called as recursive function.
↳ Needs a specific condition to stop.

→ Stack space is used in the process of recursion.
↳ Advantages
↳ Disadvantages

• Two phases of recursive functions
↓
Winding → Unwinding → When base condition is reached to return control.
↳ function keeps calling itself till base condition.

(•) Recursive Data Structures ⇒ Stack Data Structure.
⊛ Run time activity ⊛
⊛
↳ Function called → activation created
function returned → ~~—~~ popped.
⊛

(1) Tail Recursion.
↳ recursive call is last thing executed by function.

(2) Direct Recursion → When function calls itself directly.
↳ Ref. Algorithm for factorial.

(3) In-direct Recursion → When funct° calls other function that inturn calls ~~other funct°~~. the funct°.
↳ Functions in indirect recursion ⇒ mutually recursive functions

(•) Tree Recursion ⇒ Not that imp..

(•) Difference → Recursive & Iterative programming
↳ Parameters → Requirement space
→ Speed.
↳ Solving Technique
↳ Examples.

O.S. is responsible
Storage Management
↳ Sequential Methods.
↳ First Fit
↳ Best Fit
↳ Worst Fit

⊛
Also write about the memory manager. ⊛
Responsible for memory operations.

→ The condition of splitting primary memory into segments as the memory is allocated and deallocated is known as **Fragmentation**. → Worst fit attempts to reduce this.

Types → External (Not contiguous).
Types → Internal (Unused memory).

External can be reduced by compaction/shuffling.

Internal can be reduced by efficiently assigning the smallest partition but sufficient for process

- Boundary Tag Method. → Not imp.
- Buddy Systems → Imp. [10 Marker]
- Buddy Systems → Types.

Types:
- Binary Buddy System.
  - Recurrence relation.
  - $L_i = L_{i-1} + L_{i-1}$
- Fibonacci Buddy System.
  - Recurrence relation.
  - $L_i = L_{i-1} + L_{i-2}$

- Compaction & Garbage Collection.