Asymptotic Notations

- Asymptomatic analysis of an algorithm refers to defining the mathematical boundation of its runtime performance.
- Using this we can conclude the best case, average case and worst case scenario of an algorithm.
- The time required by an algorithm falls under three types:
  - **Best case**: Minimum time required for program execution
  - **Average case**: Average time required for program execution.
  - **Worst case**: Maximum time required for program execution.
- Asymptomatic notations are the expressions that are used to represent the complexity of an algorithm.
- Types of Asymptomatic notations are:

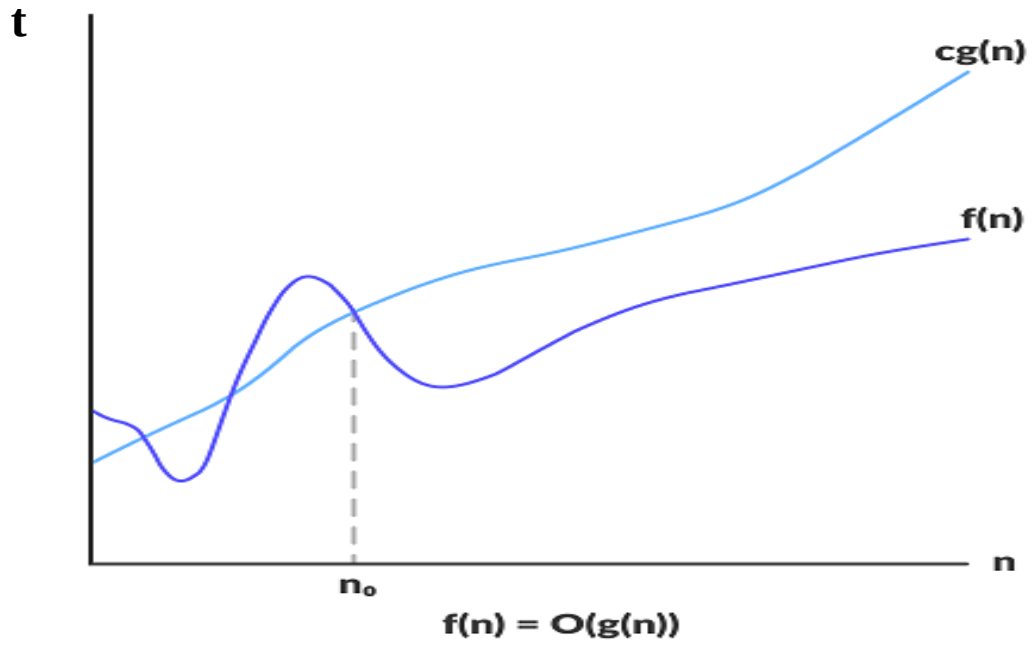
i) Big-Oh Notation (O)

ii) Big-Omega Notation(Ω)

iii) Big -Theta Notation (Θ)

Asymptotic Notations...

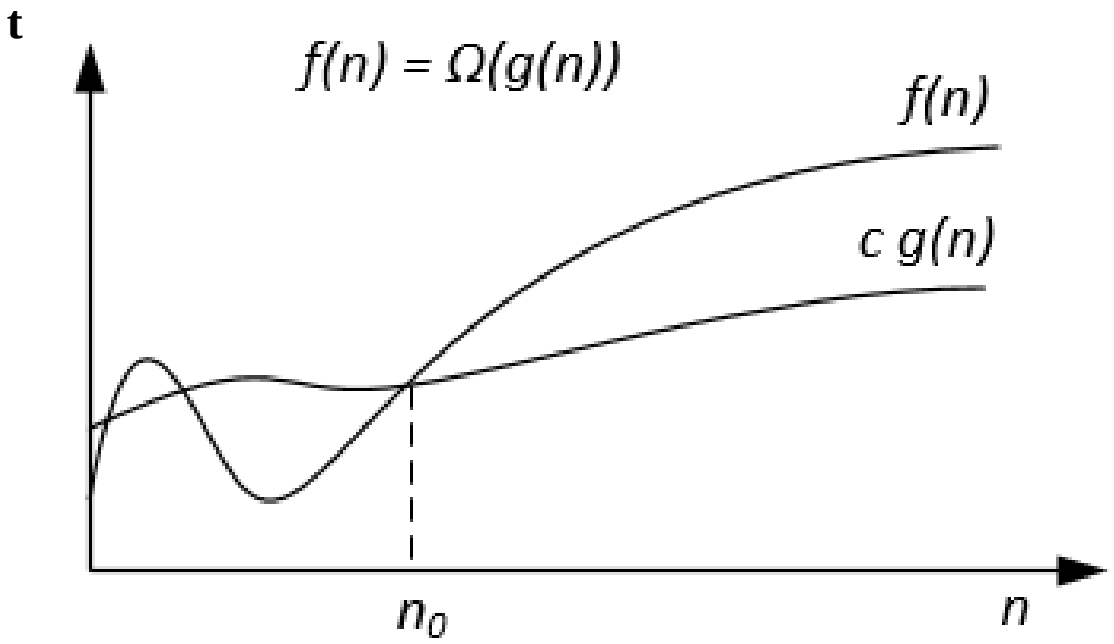
- **Big-Oh Notation (O)**
  - It describes the worst case scenerio,it represents the upper bound running time complexity of an algorithm .



  - if $f(n) \leq cg(n)\ \forall\ n \geq n_0$ where $c>0$ and $n_0 >=1$ then we can say that $f(n) = O(g(n))$.

Asymptotic Notations...

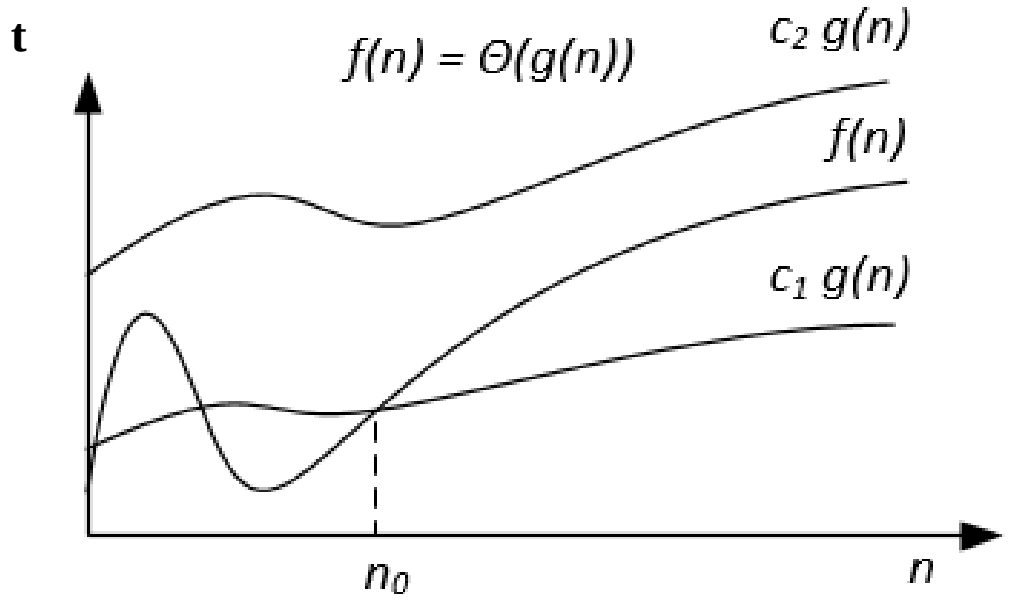
- **Big-Omega Notation(Ω)**
  - It describes the best case scenerio,it represents the lower bound running time complexity of an algorithm .



  - if f(n)>=c.g(n) ∀ n≥n_0 where c>0 and n_0 >=1 then we can say that f(n) = **Ω** (g(n)) .

Asymptotic Notations...

- **Big -Theta Notation (Θ)**
  - It describes the average case scenerio,it represents the lower bound and upper bound of an algorithm .



  - if c1g(n)<=f(n)<=c2.g(n) ∀ n≥n_0 where c1,c2>0 and n_0 >=1 then we can say that f(n) = **Θ** (g(n)) .

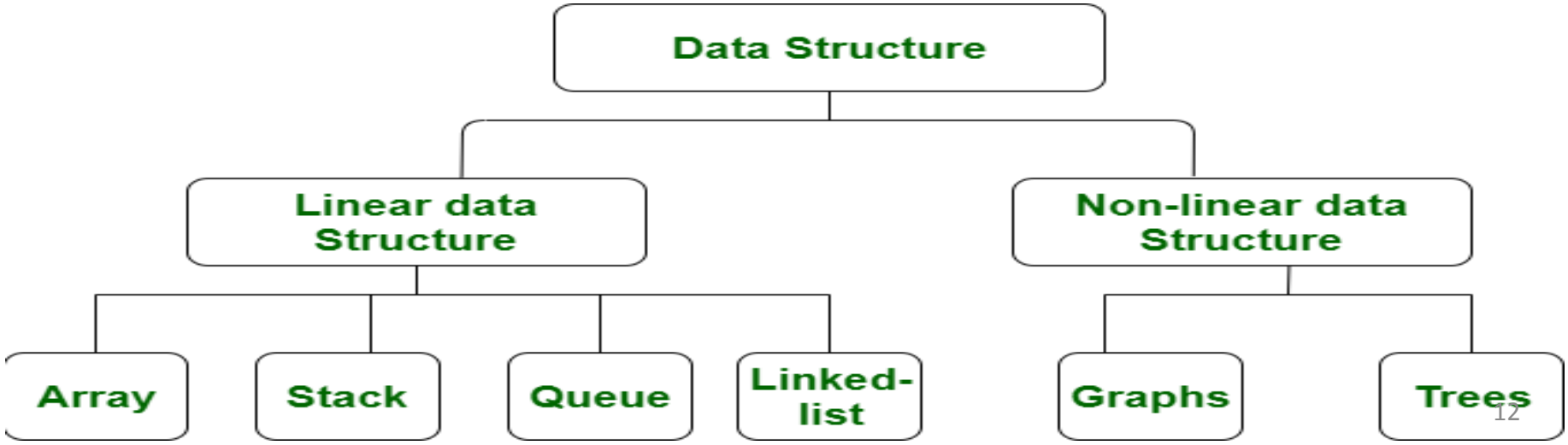
Linear & Non-linear Data Structure

➢Linear Data Structure:

- ➢ In this data elements are arranged sequentially or in linear fashion.
- ➢ It involves single level ,therefore we can traverse all the elements in single run only.
- ➢ Linear data structures are easy to implement because computer memory is arranged in a linear way.

➢Non -Linear Data Structure:

- ➢In this data elements are not arranged sequentially or linearly .
- ➢It does not involve single level, therefore, we can't traverse all the elements in single run.
- ➢ They are not easy to implement in comparison to linear data structure.
- ➢ It utilizes computer memory efficiently in comparison to a linear data structure.



Linear & Non-linear Data Structure....

- **Difference between Linear and Non-Linear Data structures :**

Sr. No.	Key	Linear Data Structures	Non-linear Data Structures
1	Data Element Arrangement	In linear data structure, data elements are sequentially connected and each element is traversable through a single run.	In non-linear data structure, data elements are hierarchically connected and are present at various levels.
2	Levels	In linear data structure, all data elements are present at a single level.	In non-linear data structure, data elements are present at multiple levels.
3	Implementation complexity	Linear data structures are easier to implement.	Non-linear data structures are difficult to understand and implement as compared to linear data structures.
4	Traversal	Linear data structures can be traversed completely in a single run.	Non-linear data structures are not easy to traverse and needs multiple runs to be traversed completely.
5	Memory utilization	Linear data structures are not very memory friendly and are not utilizing memory efficiently.	Non-linear data structures uses memory very efficiently.
6	Examples	Array, List, Queue, Stack.	Graph,Tree.