

ASYMPTOTIC NOTATION AND ARRAYS

BY

MRS. B. GOMATHI
ASSISTANT PROFESSOR



ASYMPTOTIC NOTATIONS

- Asymptotic notations are used to represent the complexities of algorithms for asymptotic analysis.
- These notations are mathematical tools to represent the complexities.
- There are three notations that are commonly used.

Big Oh Notation

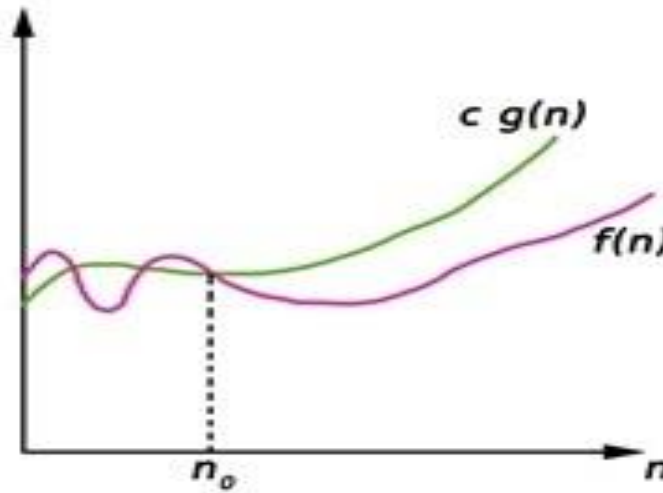
Big Omega Notation

Big Theta Notation



BIG OH NOTATION

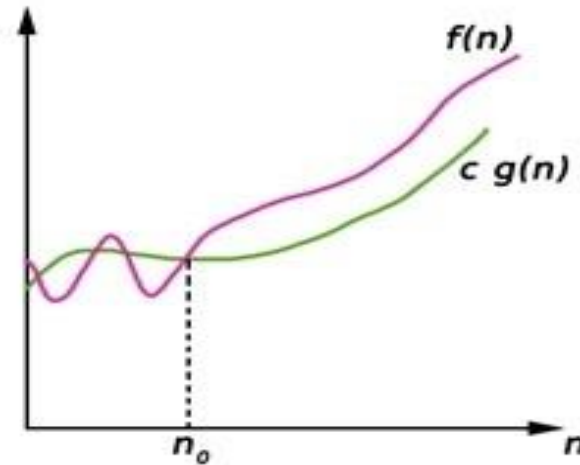
- Big-Oh (O) notation gives an upper bound for a function $f(n)$ to within a constant factor.



- We write $f(n) = O(g(n))$, If there are positive constants n_0 and c such that, to the right of n_0 the $f(n)$ always lies on or below $c * g(n)$.
- $O(g(n)) = \{ f(n) : \text{There exist positive constant } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq c g(n), \text{ for all } n \geq n_0 \}$

BIG OMEGA NOTATION

- Big-Omega (Ω) notation gives a lower bound for a function $f(n)$ to within a constant factor.

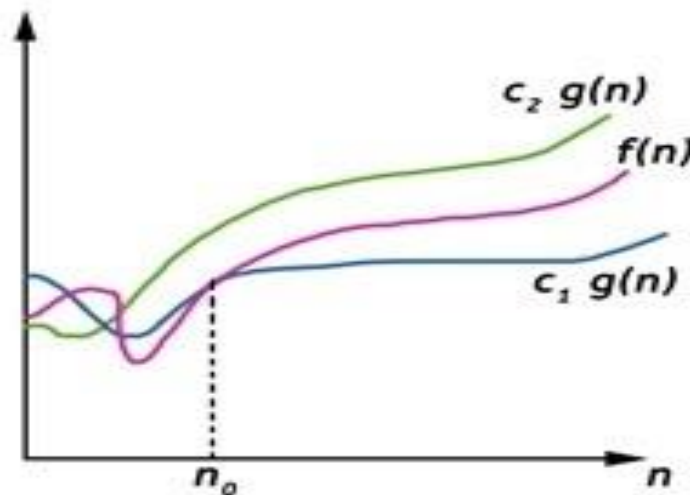


- We write $f(n) = \Omega(g(n))$, If there are positive constants n_0 and c such that, to the right of n_0 the $f(n)$ always lies on or above $c \cdot g(n)$.
- $\Omega(g(n)) = \{ f(n) : \text{There exist positive constant } c \text{ and } n_0 \text{ such that } 0 \leq c g(n) \leq f(n), \text{ for all } n \geq n_0 \}$



BIG THETA NOTATION

- Big-Theta(Θ) notation gives bound for a function $f(n)$ to within a constant factor.



- We write $f(n) = \Theta(g(n))$, If there are positive constants n_0 and c_1 and c_2 such that, to the right of n_0 the $f(n)$ always lies between $c_1 * g(n)$ and $c_2 * g(n)$ inclusive.
- $\Theta(g(n)) = \{f(n) : \text{There exist positive constant } c_1, c_2 \text{ and } n_0 \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \text{ for all } n \geq n_0\}$



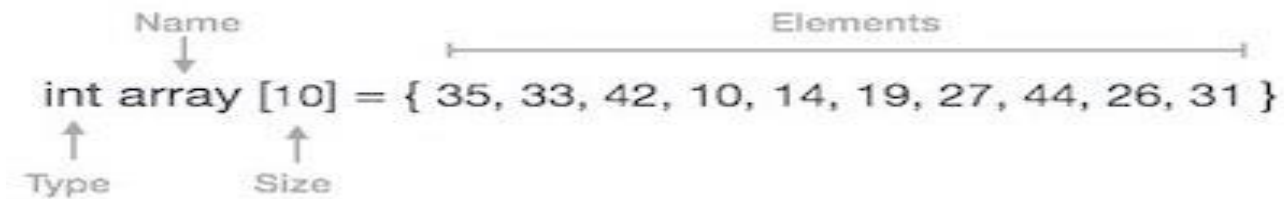
ARRAYS

- Array is a container which can hold a fix number of items and these items should be of the same type.
- Most of the data structures make use of arrays to implement their algorithms.
- **Element** – Each item stored in an array is called an element.
- **Index** – Each location of an element in an array has a numerical index, which is used to identify the element.



ARRAY REPRESENTATION

- Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.



- Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.



- As per the above illustration, following are the important points to be considered.
- Index starts with 0.



BASIC OPERATIONS

Following are the basic operations supported by an array.

- **Traverse** – print all the array elements one by one.
- **Insertion** – Adds an element at the given index.
- **Deletion** – Deletes an element at the given index.
- **Search** – Searches an element using the given index or by the value.
- **Update** – Updates an element at the given index.

