RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

Lecture : 01

Mr. Nilesh
Assistant Professor
Computer Science & Engineering

Outline

- Introduction of Design and Analysis
- Analysis of Algorithms
- Need of Analysis
  - I. Methodology of Analysis
  - II. Behavior of Algorithms
  - III. Asymptotic Notations (ASN)

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

Introduction of Design and Analysis

ALGORITHMS

- Consists of Finite set of Steps to solve a problem

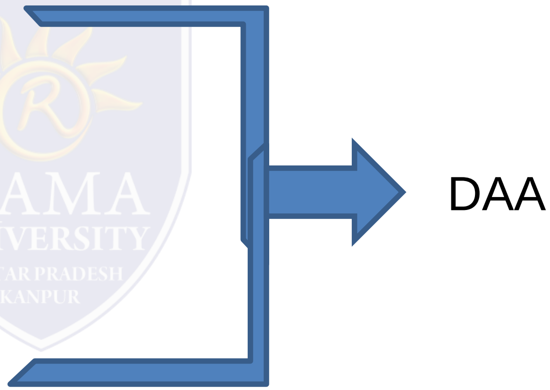
  Ex: 1. X= Y+Z
  2. Read(a)
  3. for i 1 to n
     X=Y+Z

  One or more operations
  I. definiteness(Clear)
  II. Effective
- May accept one or More Inputs
- Must produce at least on Output

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

❑ Life Cycle Steps

1. Problem Define
2. Requirements Specification
3. Design(Logic)
4. Express(Algo/Flowchart)
5. Validation(correctness)
6. Analysis
7. Implementation
8. Testing & Debugging



RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

❑ Need of Algorithms

- ❖ Resource Comparison
  1. Time
  2. Space
  3. Energy
  4. Bond width
  5. Register

- ❖ Performance Comparison : Which also is best among all.

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

❑ Methodology of Analysis

How much time taken by it

X=Y+Z

It depends on- Platform

- Hardware
- Software

1. Posterior Analysis:
   I. Platform Dependent
   II. Experimentation
2. Priori Analysis:
   I. Platform Independent
   II. Order of Magnitude
   III. Operations: +,-,*

❑ Need of Algorithms

❖ Posteriori Analysis

Advantages: Exact/ Real values of time & Space in units

Disadvantage: Difficult to carry out

Non- Uniform values gives.

RAMA
UNIVERSITY
UTTAR PRADESH
KANPUR

❖ Oder of Magnitude: refers of frequency (No. of times) of the function/operations Involved in the steps/ statements.

Behavior of Algorithms

- Types of Analysis:
  1. Worst case
  2. Best case
  3. Average Case

- Consider the Example

Given ---→ (a1, a2, a3, a4 … an)

I/P ---→ Inc. Order, I1
Dec. Order, I2
Random ,I3

RAMA UNIVERSITY UTTAR PRADESH KANPUR

Behavior of Algorithms

- Worst-case: maximum number of steps taken on any instance of size n.
- Best-case: minimum number of steps taken on any instance of size n.
- Average case: average number of steps taken on any instance of size n.

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

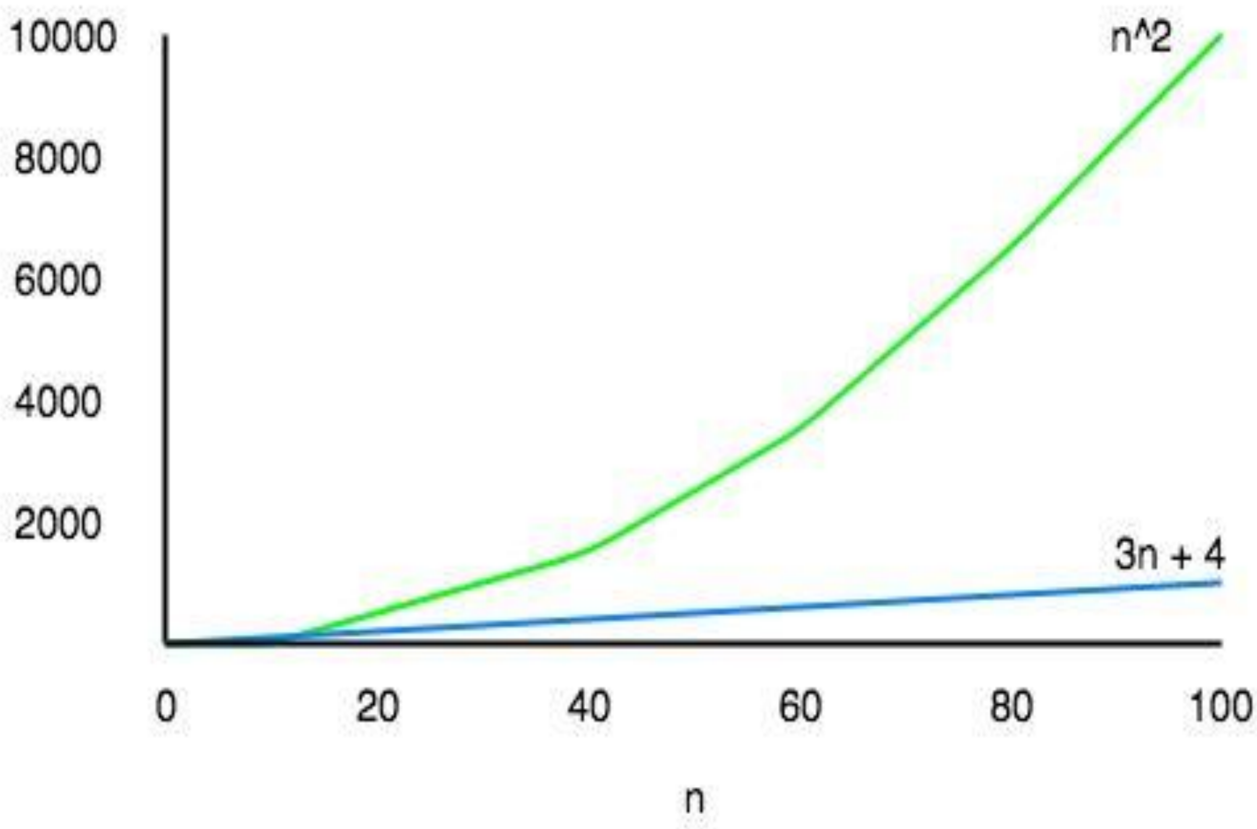
Asymptotic Notations

- We can never provide an exact number to define the time required and the space required by the algorithm.
- It is an express or standard notations of time and space, known as Asymptotic Notations.
- Mathematical Representation
- Bounds of Function
  1. Upper
  2. Lower
  3. Tight

RAMA UNIVERSITY UTTAR PRADESH KANPUR

❑ Asymptotic Notations

- Time complexity of $T(n) = (n^2 + 3n + 4)$ → quadratic equation
- For large values of n, the 3n + 4 part will become insignificant compared to the n2 part.



Asymptotic Notations

- Types of Asymptotic Notations

  We use three types of asymptotic notations to represent the growth of any algorithm, as input increases:

  1. Big Theta (Θ)
  2. Big Oh(O)
  3. Big Omega (Ω)

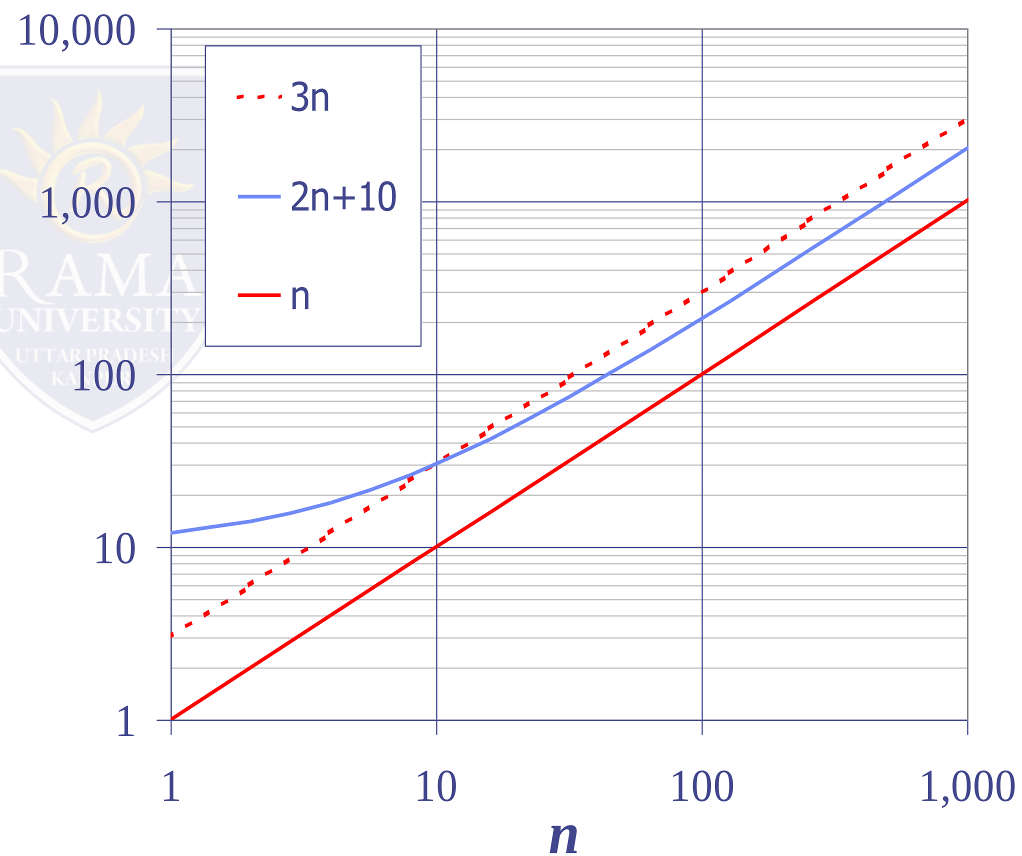
RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

❑ Big Oh(O)

- Given functions $f(n)$ and $g(n)$, we say that $f(n)$ is $O(g(n))$ if there are positive constants c and n_0 such that

  $f(n) \leq cg(n)$ for $n \geq n_0$

- Example: $2n + 10$ is $O(n)$
  - $2n + 10 \leq cn$
  - $(c - 2)\, n \geq 10$
  - $n \geq 10/(c - 2)$
  - Pick $c = 3$ and $n_0 = 10$



❑ Big Omega (Ω) & Big- Theta

→big-Omega

- $f(n)$ is $\Omega(g(n))$ if there is a constant $c > 0$
- and an integer constant $n_0 \geq 1$ such that
- $f(n) \geq c \bullet g(n)$ for $n \geq n_0$

→big-Theta

- $f(n)$ is $\Theta(g(n))$ if there are constants $c' > 0$ and $c'' > 0$ and an integer constant $n_0 \geq 1$ such that $c' \bullet g(n) \leq f(n) \leq c'' \bullet g(n)$ for $n \geq n_0$

RAMA UNIVERSITY UTTAR PRADESH KANPUR

❑ Notes

→Big-Oh

- f(n) is O(g(n)) if f(n) is asymptotically less than or equal to g(n)

→big-Omega

- f(n) is Ω(g(n)) if f(n) is asymptotically greater than or equal to g(n)

→big-Theta

- f(n) is Θ(g(n)) if f(n) is asymptotically equal to g(n)

Small/Little Notation:

- Small- Oh: Proper Upper Bond
- Small-Omega: Proper Lower Bond

❑ Some common asymptotic notations –

constant	–	$O(1)$
logarithmic	–	$O(\log n)$
linear	–	$O(n)$
n log n	–	$O(n \log n)$
quadratic	–	$O(n^2)$
cubic	–	$O(n^3)$
polynomial	–	$n^{O(1)}$
exponential	–	$2^{O(n)}$

RAMA UNIVERSITY
UTTAR PRADESH
KANPUR

❑ Practice

Q: What is time complexity of fun()?

```
int fun(int n)
{
  int count = 0;
  for (int i = n; i > 0; i /= 2)
    for (int j = 0; j < i; j++)
      count += 1;
  return count;
}
```

Options:

A. O(n^2)
B. O(nLogn)
C. O(n)
D. O(nLognLogn)

❑ Practice

Q: What is the time complexity of fun()?

```
int fun(int n)
{
 int count = 0;
 for (int i = 0; i < n; i++)
   for (int j = i; j > 0; j--)
     count = count + 1;
 return count;
}
```

Options:

A. Theta (n)
B. Theta (n^2)
C. Theta (n*Logn)
D. Theta (nLognLogn)

RAMA UNIVERSITY UTTAR PRADESH KANPUR