Data Structure Using C

Ch. 01 Algorithm Analysis

Ch.1 Algorithm Analysis

- Topics
  - The analysis of algorithm.
  - Time and space complexities.
  - Asymptotic notation.
  - Classes of algorithm.
  - Big-Oh Notation
  - Big-Omega Notation
- Marks : 5

Introduction To Algorithm

- To work with a computer we have to use a computer program.
- To write a computer we have to provide step by step command to perform a task.
- **What is Algorithm?**
  - An algorithm is just a detailed sequence of simple steps that are needed to solve a problem.

Introduction To Algorithm

- Features of Algorithms:
  - Finiteness : (મર્યાદિત)
    - The algorithm must terminate after the finite steps.
  - Definiteness : (નિશ્ચિત)
    - Each step must be clear.
  - Effectiveness : (અસરકારકતા)
    - Each step must be effective
  - Input :
    - There can be zero of more input.
  - Output :
    - There should be at lease one output

Introduction To Algorithm

- There are three types of Logic structures are possible in ALGORITHM.
  - Sequence Logic.
    - In this logic structure all the instructions are written in order to performed.
  - Decision Logic
    - When more then one option is available then we can use decision logic with.

Introduction To Algorithm

- Looping Logic :
  - When one or more instructions may be executed several times then we can use looping logic.

Example Of Sequence Logic Algorithm

Write an algorithm to read and print value.

Step-1 Input the number

Step-2 Print the number

Step-3 Stop

Example Of Decision Logic Algorithm

Write an algorithm to find the maximum number from given two number.

Step-1 Input value of no1 and no2

Step-2 If no1>no2, if yes then print no1 as maximum else no2 as maximum

Step-3 Stop

Example Of Looping Logic Algorithm

Write an algorithm to print 1 to 10 serial number.

Step-1 Input i=1

Step-2 Check is i<=10, if yes then goto step-3 else goto Step-5

Step-3 Print I

Step-4 i=i+1, goto Step-2

Setp-5 Stop

Introduction To Algorithm

- Algorithm to goto home by : Taxi

  Step-1 Go to the taxi stand

  Step-2 Get in a Taxi

  Step-3 Give address to taxi driver.

- Algorithm to goto home by : Rixo

  Step-1 Go to rixo stand

  Step-2 Get a Rixo

  Step-3 Give address to rixo driver.

Introduction To Algorithm

- Algorithm to goto home by : Bus

  Step-1 Go to the City Bus Stand

  Step-2 Get a Bus number 7

  Step-3 Get off on Trikon bag

  Setp-4 Walk two street north to my home

  - All of these algorithms accomplish exactly the same goal, but each algorithm does it in completely different way.
  - All above algorithm has different cost and travel time.
  - Taxi will fastest way but costly too.
  - Bus will less expensive but lot slower.

Analysis Of Algorithm

- In computer science, the **analysis of algorithms** is the determination of the number of resources necessary to execute them.
- Most algorithms are designed to work with inputs of undefined length.
- Algorithm analysis is an important part of a wide computational complexity theory which provides theoretical estimates for the resources needed by any algorithm which solves a given computational problem.

Analysis Of Algorithm

- Cost Models of Algorithm Analysis :
  - Two cost models are generally used to algorithm analysis.
    - Uniform cost model
    - Logarithmic cost model

Analysis Of Algorithm

- Uniform cost model
  - The uniform cost model is also called as uniform-cost measurement.
  - It assigns a constant cost to every machine operation regardless (ધ્યાન આપ્યા વિના) of the size of the numbers involved.

Analysis Of Algorithm

- Logarithmic cost model
  - The logarithmic cost model is also called as logarithmic-cost measurement.
  - It assigns a cost to every machine operation proportional (પ્રમાણસરનું) to the number of bits involved.

TIME Complexity

- A program executes number of instructions during the execution time is called as its **Time Complexity**.
- The time complexity of an algorithm is commonly expressed using big O notation, which excludes coefficients (ગુણક સંખ્યા) and lower order terms.

Space Complexity

- Space Complexity of an algorithm is total space taken by the algorithm with respect to the input size.
- Space complexity includes both Auxiliary (સહાયક) space and space used by input.

Asymptotic Notations
(અનંત સ્પર્શી સંકેતલિપી)

- **Asymptotic Notation** is often used to describe how the size of the input data affects an algorithm's usage of computational(ગણતરીના) resources.
- In order to choose the best algorithm for a particular task, we need to be able to judge how long a particular solution will take time to run a program.

Type & Classes of Algorithms

- There are most used algorithms are as given.
  - Simple recursive algorithms
  - Backtracking algorithms
  - Divide-and-conquer algorithms
  - Dynamic programming algorithms
  - Greedy algorithms
  - Branch-and-bound algorithm
  - Brute force algorithms

Simple recursive algorithms :

- A simple recursive algorithm can,
  - Solve the base cases directly
  - Recurs with a simpler sub-problem
  - Does some extra work to convert the solution to the simpler sub-problem.

Simple recursive algorithms :

- Example
  - To count the number of elements in a list,
    - If the list is empty, return zero; otherwise
    - Step past the first element, and count the remaining elements in the list.
    - Add one to the result

Backtracking Algorithms

- A backtracking algorithm is based on a depth-first recursive search.
  - Tests to see if a solution has been found, returns it; otherwise
  - For each choice that can be made at this point,
    - Make that Choice
    - Recur
    - If the recursion returns a solution, return it
  - If no choices remain, return failure

Divide and Conquer Algorithms:

- A divide and conquer algorithm consists of two parts.
  - Divide the problem into smaller sub-problems of the same type and solve these sub-problems recursively
  - Combine the solutions to the sub-problems into a solution to the original problem
- Traditionally, an algorithm is only called divide-and-conquer if it contains two or more recursive calls.

Dynamic programming algorithms

- A dynamic programming algorithm remembers past results and uses them to find new results.
- Dynamic programming in generally used for optimization problems in which
  - Multiple solutions exist, and need to find the best one.
  - Requires optimal substructure and overlapping sub-problem.

Dynamic programming algorithms

- Optimal substructure :Optimal solution contains optimal solutions to subproblems
- Overlapping subproblems: Solutions to subproblems can be stored and reused in a bottom-up fashion

Greedy Algorithms :

- A greedy algorithm sometimes works well for optimization(સારા પરિણામ) problems.
- A greedy algorithm works in phases.
  - You take the best you can get right now, without regard for future consequences(પરિણામ).
  - You hope that by choosing a local optimum(વધુ અનુકૂળ પરિસ્થિતિ) at each step, you will end up at a global optimum.

Branch-and-bound algorithms:

- Branch-and-bound algorithms are generally used for optimization problems.
- As the algorithm progresses, a tree of sub-problems is formed.
- The original problem is considered the root problem.
- A method is used to construct an upper and lower bound for a given problem.

Brute force algorithms :

- A brute force algorithm simply tries all possibilities until a satisfactory solution is found.
- Such an algorithm can be
  - **Optimizing :** Find the best solution.
  - This may require finding all solutions, or if a value for the best solution is known, it may stop when any best solution is found.
  - Satisfying : Stop as soon as a solution is found that is good enough.

Big OH Notations

- **Big OH notation** also called **Landau's symbol**, is a symbolism used in complexity theory, computer science, and mathematics to describe the asymptotic (અનંત સ્પર્શક) behavior of functions.
- Basically, it tells you how fast a function grows or declines(ઘટવું).

Big OH Notations

- *Landau's symbol* comes from the name of the German number theoretician Edmund Landau who invented the notation.
- The letter O is used because the rate of growth of a function is also called its ***order*.**

Big OMEGA Notation

- To describe lower bounds we use the big-omega notations.
  - f(n)=O(g(n)) usually defined by saying for some constant c>0 and all large enough.

Habituate

- What is Algorithm?
  - An algorithm is just a detailed sequence of simple steps that are needed to solve a problem.
- Features of Algorithms:
  - Finiteness : (મર્યાદિત)
  - Definiteness : (નિશ્ચિત)
  - Effectiveness : (અસરકારકતા)
  - Input
  - Output
- Algorithms Logics :
  - Sequence Logic
  - Decision Logic
  - Looping Logic

Habituate

- Analysis of algorithms
  - Analysis of algorithms is the determination of the number of resources necessary to execute them.
- Cost Models of Algorithm Analysis :

  1. Uniform cost model **2.** Logarithmic cost model
- Time Complexity
  - A program executes number of instructions during the execution time is called Time Complexity.
- Space Complexity
  - Total space taken by an algorithm with respect to the input size is called Space Complexity.
- Asymptotic Notation
  - Asymptotic Notation is used to describe how the size of the input data affects an algorithm’s usage.

Habituate

- Type & Classes of Algorithms
  - Simple recursive algorithms
  - Backtracking algorithms
  - Divide-and-conquer algorithms
  - Dynamic programming algorithms
  - Greedy algorithms
  - Branch-and-bound algorithm
  - Brute force algorithms
- Big OH notation
  - Big OH notation is a symbolism used in complexity theory, computer science, and mathematics to describe the asymptotic (અનંત સ્પર્શક) behavior of functions.