Objectives:

- Introduction of Design and Analysis of Algorithms
- Course Blow-up
- Algorithm and Program
- Characteristics of Algorithm
- How to write and analyse an algorithm?
- Examples to write and analyse algorithm.

Design and Analysis of algorithms:

For a given problem, how we will solve on computer. For many of the problems it is relatively easy to solve. However some cleverness is needed and design algorithms which give answer very quickly. This will be the major challenge of this course.

Main goal is to design algorithm which are fast.

Like designing anything be it a computer, be it transportation vehicle, and be it a wearable thing is an art. In some sense you have to be creative and it can't be taught.

And some other sense they are very well designed. Many techniques have evolved for this purpose. Goal of this course is to study these techniques.

Prerequisites for this course

1. Familiar with some programming
2. Data structures
3. Familiar with Discrete mathematics

Approach for this course:

1. Build a Mathematical model
2. Design algorithms and study their properties on this mathematical model
3. Prove properties, prove facts about time taken.
4. Study techniques for designing fast algorithms
5. Divide and Conquer, backtracking, branch and bound algorithms
6. Problems of optimization (Greedy algorithms, Dynamic programming), graph theory and some others.
7. String matching algorithms
8. Complexity classes

Text Books:

1. Cormen H. T., Leiserson E. C., Rivest L. R., and Stein C., Introduction to Algorithms, MIT Press (2009) 3rd ed.

2. Horwitz E., Sahni S., Rajasekaran S., Fundamentals of Computers Algorithms, Universities Press (2008) 2nd ed.

Reference Books:

1. Levitin A., Introduction to the design and analysis of algorithms, Pearson Education (2008) 2nd ed.

2. Aho A.V., Hopcraft J. E., Dulman J. D., The Design and Analysis of Computer Algorithms, Addsion Wesley (1974) 1st ed.

3. Sedgewick R. and Wayne K., Algorithms, Addison-Wesley Professional (2011), 4th ed

Algorithm:

1. Step by step procedure for solving a computational problem.
2. Algorithm is written at designed time.
3. Algorithm is written in any language like English and its better if use mathematical notation.
4. Analyse the algorithm, to study whether we are achieving our objectives or in terms of time and space analysis.

Program:

1. Step by step instruction for solving a computational problem written in a specific language such as c, c++, java etc.
2. Written at implementation time.
3. Depends on H/W and OS such as Linux, windows etc. that is environment is different.
4. Analyse during testing.

Properties/characteristics of an algorithm:

1. Input: algorithms must take some input i.e. 0 or more input.
2. Output: algorithm must generate some output, without O/P algorithm does not make any sense. Eg. A function either containing parameters or not, it always return something.
3. Definiteness: every statement in the algorithm has some definiteness.
4. Finiteness: algorithm must terminate at some point, i.e. duration of algorithm must be finite.
5. Effectiveness: implies don't write any unnecessary statements. The statements that we mention in the algorithm have some purpose.

How to write and analyse an algorithm?

There is no fix syntax for writing an algorithm. Eg. Algorithm for swapping of two numbers:

$$
\begin{array}{lll}
Algo\ Swap(x,y) & Algo\ Swap(x,y) & Algo\ Swap(x,y) \\
\{ & \{ & \{ \\
temp = x; & temp \leftarrow x; & temp := x; \\
x = y; & x \leftarrow y; & x := y; \\
y = temp; & y \leftarrow temp; & y := temp; \\
\} & \} & \}
\end{array}
$$

- No data type is required when writing an algorithm.
- Extra variables are not declared in the algorithm.

Criteria to analyse the algorithm:

Time: implies how much time efficient our algorithm is. We must analyse how much time it is taking. Time of the algorithm is measured in the form of function.

Space: as algorithm is transformed into a program and its execution is on machine, so we must know how much space is required.

Other parameters CPU registers, power consumption and data transfer rate are also considered while designing an algorithm.

Example 1:

$$Algo\ Swap(x, y)$$

$$\{$$

$$temp = x; \text{------}(1)$$

$$x = y; \text{---------}(1)$$

$$y = temp; \text{------}(1)$$

$$\}$$

$$Total\ \ time : f(n) = 3 = cons\tan t\ \ time$$

$$= O(1)$$

$$Similarly\ \ Space = O(1)$$

Example 2:

$$Algo\ \ Add(X,n)$$

$$\{\ sum = 0;\text{----------}(1)$$

$$for(j = 0;\ j < n;\ j++)\text{-------}(n+1)$$

$$\{$$

$$sum = sum + X[j];\text{---------}(n)$$

$$\}$$

$$return\ \ sum;\text{----------}(1)$$

$$\}$$

$$Total\ \ time : f(n) = 2n+3$$

$$= O(n)$$

$$Similarly\ \ Space :\ \ X\text{---}(n\ \ words),$$

$$n\text{---}(1word),sum : (1word),\quad j\text{---}(1word)$$

$$totalspace : n+3 = O(n)$$

Example 3

$$Algo\ Product(A,B,C,n)$$

```
{
for(i = 0; i < n; i++)-------(n+1)
{
for(j = 0; j < n; j++)-------n×(n+1)
{    C[i,j] = 0 ----------(n×n)
   for(k = 0; k < n; k++)-------n×n×(n+1)
   {
  C[i,j] = C[i,j] + A[i,k] * B[k,j] -----(n×n×n)
   }

}
}
}
```

$$Total\ time: f(n) = 2n^3 + 3n^2 + 2n + 1$$

$$= O(n^3)$$

$$Similarly\ Space: \quad A,B,C ---(n^2\ words),$$

$$n ---(1\,word), i, j, k ---(1\,word)$$

$$total\ Space: O(n^2)$$

THANKS !