Algorithm Complexity :-

- The term algorithm complexity measures how many steps are required by the algorithm to solve the given Problem.

- Each algorithm takes some storage space to store and time to execute.

→ These two factors measure the efficency of algorithm.

- The major problem in solving programming is not how to solve the Problem, but the Problem is how to solve the problem efficently.

[for example :-] if someone want to calculate the sum of 2,2,2,2,2.

(i) $2+2+2+2+2 = 10$

(ii) other $2*5 = 10$

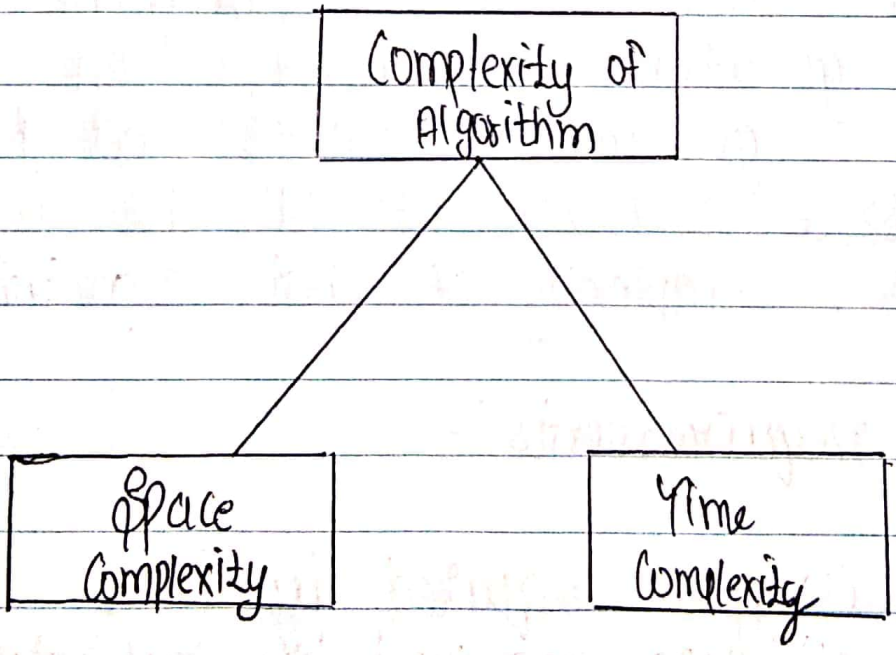
(The Problem is solved in both ways by adding five times or multiplying).

The Primary resource that an algorithm consumes are:-

Time :- Time required executing the algo.
Space :- Amount of memory used by algo

- Algorithmic Complexity :- The most important properties of an algorithm is how its execution time increases as the Problem is made large.
This is called algorithmic Complexity of algorithm.

The Complexity of an algorithm depends on two factors:-



1. Space Complexity :-

(How much memory cells are required to solve the problem).

- Space Complexity means Number of memory cells an algorithm to solve a particular problem.
- It is expressed as a function f(n) that describes the amount of memory.
- A good algorithm keeps this number as small as possible.
- If time and space requirement of the algorithm is more, then complexity of the algorithm is more and if time and space requirement of the algorithm is less, complexity of that algorithm is less

Space requirements :-

(i) Space required by data.
(ii) Space required by instructions.

(i) Space required by data :- This includes space required to store variables and constants which is generally fixed. Sometimes, space is allocated dynamically (i.e. at the runtime).

(ii) Space required by instructions :- This is the space required to store the instruction sets.

Short Definition

If the number of bits required to complete its run for an input of size (n) is always less than or equal to S(n)

$$bits \leq S(n)$$

2. Time Complexity :-

(How much time algorithm required to run the completion).

- Time Complexity of an algorithm is the amount of time it needs to run.
- The main objective of time complexity is to compare the performance of different algorithms in solving the same problem.
- Time complexity is measured in terms of input size "n".
- If the input size of algorithm is more, the complexity will be more and if input size of algorithm is less, the complexity will be less.
- To calculate the time complexity of an algorithm, the basic approach is to count the number of times, a key operation is executed.

{Short}

If the number of steps required to complete its run for an input of size n is always less than equal to T(n).

$$n \ll Tn$$

Time AND SPACE TRADE OFF :-

* A trade off is a situation where one thing increases and another thing decreases.
  It is a way to solve a Problem in:-
  - Either in less than and by using more space
  - In very little space by spending a long amount of time.

> The best algorithm for a particular problem is that it requires minimum time to complete it by taking the minimum amount of space into the memory.

But Pratically both these objectives are not possible to achieve at same time.

for example:-

→ Consider the concept of any book, in which index is prepared for the fast searching of data.

↳ we can save our time of searching the topic from the book with extra pages attached for index.

↳ It means if we are not compromise with the space that is consumed for index of book, then we have to waste some extra time to search the topic from book.

Types:-

1. Compressed or Uncompressed Data
2. Smallar code or loop unrolling
3. Re-Rendering or Stored images