1. **Algorithm**

It is a sequence of finite steps to solve a problem

Example:

Multiply()

{

1. Take 2 numbers (a,b)
2. Multiply a, b and store the result in c
3. Return (c )

}

1. **Properties of Algorithm:**

* It should terminate after finite time
* It should produce at least one output
* It is independent of any programming language i.e. instruction given to compiler
* It should be unambiguous(Deterministic) which means no difference in output for same input
  + for same input 2,3 the same output 2\*3 =6 should come even after some days.
  + If different output comes for same input, then it is non- deterministic algorithm

1. **Design Algo**

**Steps required to construct algorithm:**

1. Problem definition
2. Design algo ( Designing a solution patterns)
   1. DAC (Divide And Conquer)
   2. Dynamic Programming
   3. Greedy Technique
   4. Backtracking
   5. Branch and Bounce
3. Testing
4. Implementation
5. Analysis

1. **Data Structure:**

Any Algorithm pattern we pick, we need to store the date to design a solution. We use different structures to store the data.

Eg: Queue, Linked List, Array, Stack

Every algorithm has its own pros and cons, based on the necessity to find optimistic solution we have to pick proper set of algorithms and its respective data structures.

1. **Analysis** :

It means observation, when we have multiple solutions for a single problem we need to analyze them.

Problem has 4 solutions

Sol1 sol2 sol3 sol4

Which is best we need to analyze

One solution can be good at whereas the one can be good at something else

1. **Factors of Analysis:**

Time complexity (TC)

Space Complexity (SC)

* Depending on the use case we need to pick which solution is optimal among all 4 solns
* Sometimes there can be a tradeoff between these 2.
* sometimes we get better TC but takes more SC.
* Usually we give more priority to Time Complexity if use-case didn’t provide anything specifically.

|  |  |  |  |
| --- | --- | --- | --- |
| Analysis | Sol1 | Sol2 | Good |
| TC | O(n) | O(n) | No |
| SC | O(1) | O(n) | Yes |

In the above case , both have same TC but sol1 has better SC

1. **Types of Analysis:**A piece of paper with writing on it

   Description automatically generated with medium confidence

n-> linear time

n^2 --> quadratic time

* Time complexity is nothing but loop only.
* Greater the loops greater the time complexity ( directly proportional)
* In best case any code may get executed but in worst case only optimal code run faster