Algorithm. (1)

Finite Set of rules or instructions to be followed to solve a problem.

Input → Program → output
Algorithm

Algorithm plays a vital role in

* Computer Science - basics for implementation
* Mathematics :: Solving mathematical Problem ex Finding optimal. Solution to linear eqn.
* Operation research :: Used to optimize and make decision in various fields (resource allocation transportation)
* Data Science ::
Analyze, process & extract (marketing, Finance & Healthcare)

* Algorithm is a language dependent

Characteristics of Algorithm :

* Algorithm should be unambiguous.
  (Steps must be clear & lead to only one meaning).

* Finiteness.
  - Must terminate after finite number of steps.

* Definiteness.
  Steps must be clear & unambiguous.

* Input.
  - Should have 0 or more.

* Output.
  - Should have atleast one output.

* Effectiveness
  An algorithm must be developed by using simple & feasible operations.
  - operations performed in a reasonable time.

* Feasible.
  algorithm must be simple, generic & practical. it can be executed with available resources.

* Generality.
  - Should solve all problems of a particular class not a specific instance.

Consecutive integer

1 - Assign t = min (m, n)

2 → Divide m/t if remainder of division is 0, goto step 3 otherwise goto step 4.

3 → Divide n/t If remainder is 0 return t as answer and stop else goto step 4.

4 → decrease the value of t by 1 goto step 2

gcd (60, 24)
m n

t = min (60, 24)

1) t ⟸ 24

2) 60/24 = 12 r ≠ 0 goto step 4.

t = t - 1
t = 24 - 1
t = 23.
60/23 = 14 r ≠ 0.
60/22 = 16 r ≠ 0
60/21 = 18 r ≠ 0.
t=20 ⟶ 60/20 = 0 r=0 goto step 3.

24/20 = 4 r ≠ 0.

24/19
24/18
⋮

24/12 = 0 r = 0.
return t as ans $\boxed{t = 12}$
∴ gcd (60, 24) = 12

gcd (36, 48)
n. m

31415, 14142 = 1

Euclid ~0.08

Consecutive 0.55

Gcd.

Euclid (m, n)
// Input : m, n
// output : GCD (m, n)
while (n ≠ 0)
r ← m mod n
m ← n
n ← r.
return m.

example :-
m n
Gcd (60, 24)
formula gcd(m, n) = gcd (n, m mod n).
Gcd(60, 24) = Gcd (24, 60/24)
Gcd(24, 12)
Gcd 12, 24/12
Gcd (12, 0)
Gcd (12).
∴ Gcd(60, 24) = 12.

ex : 1 n m.
Gcd (36, 60)
Gcd (36, 60%36)
36, 24
24, 24/36.
24, 12
12, 12/24.
(12, 0)
∴ Gcd(36, 60) = 12

ex : 2.

m n

Gcd (252, 105)

Gcd (105, 252/105)

Gcd (105, 42)

Gcd (42, 42/105)

Gcd (42, 21)

Gcd (21, 21/42)

Gcd (21, 0)

∴ Gcd (252, 105) = 21.

ex No : 3.

m n.

Gcd (315, 99)

Gcd (99, 315/99)

Gcd (99, 18)

Gcd (18, 99/18).

Gcd (18, 9)

Gcd (9, 9/18)

Gcd (9, 0)

∴ Gcd (315, 99) = 9.

example problems.

1) Gcd (84, 30)

2) Gcd (198, 168)

3) Gcd (150, 100)

Space Complexity.

Fixed Space. → not depends on pgm i/p & o/p.
only depends on variable & const.

Variable space.
↳ Size depends on instance $\boxed{i}$

$$S(P) = \underbrace{c}_{\text{constant}} + \underbrace{S_P}_{\text{instant characteristics}}$$

```
Proc. Sq(n).
    return n*n.
end proc.
```

Constant

```
Proc. Sum(A,n)
    S ← 0.
    for i ← 0 to n-1 do.
        S ← S + A[i].
    end for.
end proc.
```

n, S, i takes constant
Sum of 3 units
A is array