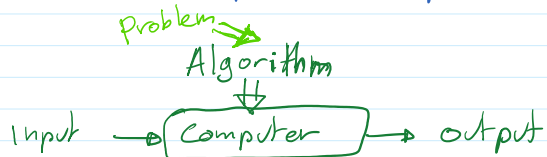


DEF: An algorithm is a sequence of unambiguous instructions for solving a problem.

solving.- to obtain an required output from a specified input.



Important points:

- The non-ambiguity of each step cannot be compromised.
- The range of the input must be specified carefully.
- The same algorithm can have multiple representations.
- There may be more than one algorithm to solve the same problem.
- Algorithms for the same problem based on different ideas can have widely different characteristics.
  - differences on resource needs.

E.G.

Greatest Common Denominator:

$\gcd(m, n)$ : The biggest number that divides both  $m$  and  $n$ .

# Euclid's Algorithm "The Elements"

- Form 1: if  $n$  is equal to 0, answer  $m$  and stop.
- Divide  $m$  by  $n$  and assign the result to  $r$ .
  - Assign  $n$  to  $m$  and  $r$  to  $n$ .
  - Repeat from step 1.

Form 2: `euclid(m, n)`

```

while n ≠ 0 do
  r ← m mod n
  m ← n
  n ← r
return m.
    
```

← assignment

$C: = = = *$   
 $\wedge$  &

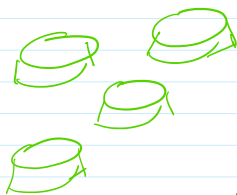
&& ||  
 and or

$= = = =$

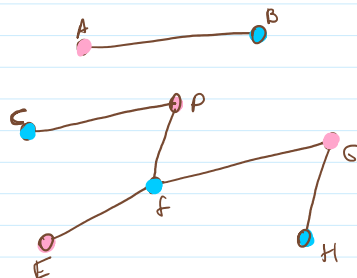
- The steps for Algorithmic Problem Solving:

## 1. Understand the Problem

- assumptions: "given an array of numbers"
- ignoring: "in the array a number appears only once"  
 "numbers range from 0 to 256"
- abstraction: Mob weddings



some guests cannot sit together.



APGE  
 BCFH

## 2. Decide on Computational needs, constraints.

- RAM Model.
  - sequential - one instruction at a time
  - Memory with random access
- PRAM
- Quantum

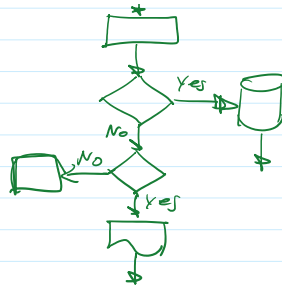
- exact vs approx solution.



- exact vs approx solution.

### 3 - Specify Algorithm

- Pseudocode.



### 4 - Prove Correctness

- Empirical testing.
- Rigorous Arguments.
  - output is correct for any valid input
  - the algorithm terminates.

### 5 - Analyse Algorithm.

- resource utilization
  - time (number of steps)
  - space (size of memory needed)
- use of randomness,
- "Stability"

## Important Problem Types:

### Sorting.

re-arranging a list in non-decreasing order.

### Searching.

find an item in a collection  
 - representation of the collection.

### String Processing.

find a sub-string.
 

- text processing.
- search data
- Genetic Data

### Graph Problems.

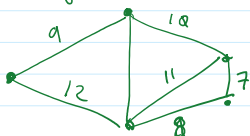
Graph:  $G = \langle V, E \rangle$



- traversals
- path.
- cycles
- adding/removing edges.

### (TSP) Travelling Salesman Problem

Weighted Graph.



find a way to visit all nodes in the graph, and return to the starting point, of minimum cost.



## Combinatorial Problems.

finding a permutation or combination of elements that satisfies certain criteria, or minimize or maximize a property.

(Knapsack)



$$\binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5}$$

5 1

These problems are HARD:

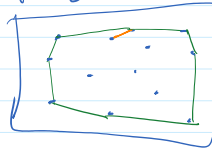
- No efficient algorithms are known.
- No proof that an efficient algorithm does not exist.

∴  $P = NP$ ?

## Geometric Problems.

Problems on points, lines, polygons.

- closest pair
- convex hull.



- line-polygon intersection.



## Numerical Problems.

- (calculus)
  - solving equations / systems of eq
  - computing integrals / roots of polynomials
  - tension networks.
- approx solutions only.

-EOF-