Data Structures.

Abstract data type.

- Combination of data structure and their operation are known as Abstract Data type.

Algorithm

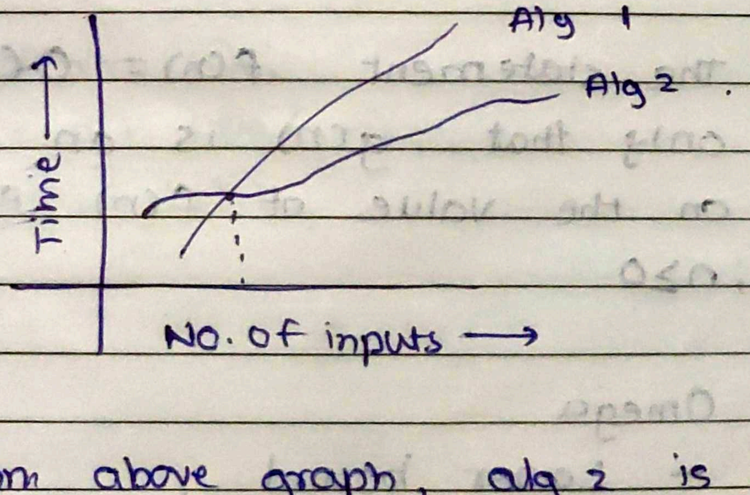
- Step by step instructions to solve a problem.

- Rate of growth

The rate at which the running time increases as a function of input is called rate of growth.

$f(n) = n^4 + 2n^2 + 100n + 500$

$f(n) = n^4$ for some $n > n_o$.



From above graph, alg 2 is better.

$\log n < \sqrt{n} < n < n\log n < n^2 < 2^n$

- Types of analysis of algoriths :
  - worst case
  - Average case
  - Best case.

- Asymptotic Notations
  1. Big O
  2. Omega Ω
  3. Theta Θ

1. Big O notation

- $O(g(n)) = f(n)$

  There exist positive constants c and n_0 such that $0 \le f(n) \le c^* g(n)$ for all $n \ge n_0$.

- The statement $f(n) = O(g(n))$ states only that $g(n)$ is an upper bou on the value of $f(n)$ for all n ; $n \ge 0$

2. Omega
   - Lower bound

3. Theta.
   - consider both upper as well as lower bound

Array

Array takes constant time to access

Recursion and Backtracking

When a function calls itself in a code, it is known as recursive

ex:
```
main ()
{   int k
    k = fun (3) ;
    printf ("%d" k);
}
  int fun (int a)
{
    int s
  if (a==1)
    return(a);
  s = a + fun (a-1);
   return (s);

 }
```

—★—

Big O notation common rules

1. Multiplicative constants can be omitted

$$7n^3 = O(n^3)$$

2. $n^a < n^b$ for $0 < a < b$.

$$n = O(n^2), \quad \sqrt{n} = O(n)$$

3. $n^a < n^b$ $(a > 0, b > 1)$

$$n^5 = O(\sqrt{2}^{\,n}), \quad n^{100} = O(1.1^n)$$

4. $(\log n)^a < n^b$ $(a, b > 0)$

$$(\log n)^3 = O(\sqrt{n})$$
$$n \log n = O(n^2)$$

5. Smaller terms can be manipulated

$$n^2 + n = O(n^2) \qquad n^5 + 1.1^n = O(1.1^n)$$

Levels of designing algorithm

- Naive algorithm.
- Algorithm by way of standard tools.
- Optimized algorithm.
- Magic algorithm.

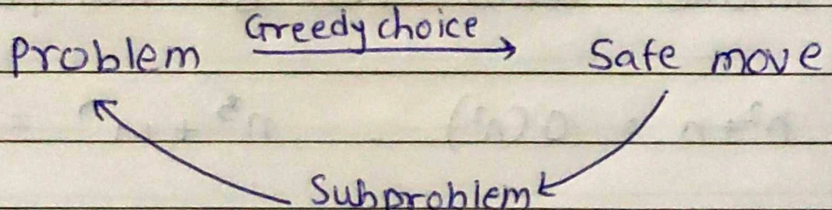
Greedy Algorithms

· main ingredients

1) safe move.
2) Prove safety.
3) solve subproblem.
4) Estimate running time.

· Greedy move can be faster after sorting.

General method :→


Problem
Greedy choice
Safe move
Subproblem