

# Content

- Complexity Theory

## Introduction

Complexity theory defines a methodology to analyse the computational complexity of crypto algorithms.

# Definitions

Information Theory, (Except One-Time Pad)  
say that all cryptosystems are breakable.

Complexity Theory, explains the  
cryptosystems which are breakable or  
unbreakable without any dependency to  
exist computational powers.

# Algorithmic Complexity :

The power of crypto; define by the required computational power to break it.

Argument #1 : Time -  $T$

( required time to break it),

Argument #2 : Space -  $S$

(required memory to break it)

## Algorithmic Complexity :

Power of Cryptosystem is defined by **big O** .

**Big O notation:** Computational Complexity degree (as an upper bound).

# Complexity Function

If the time complexity of an algorithm is defined by

$$3n^3+5n+23$$

Then ; its computational complexity is


$$O(n^3)$$

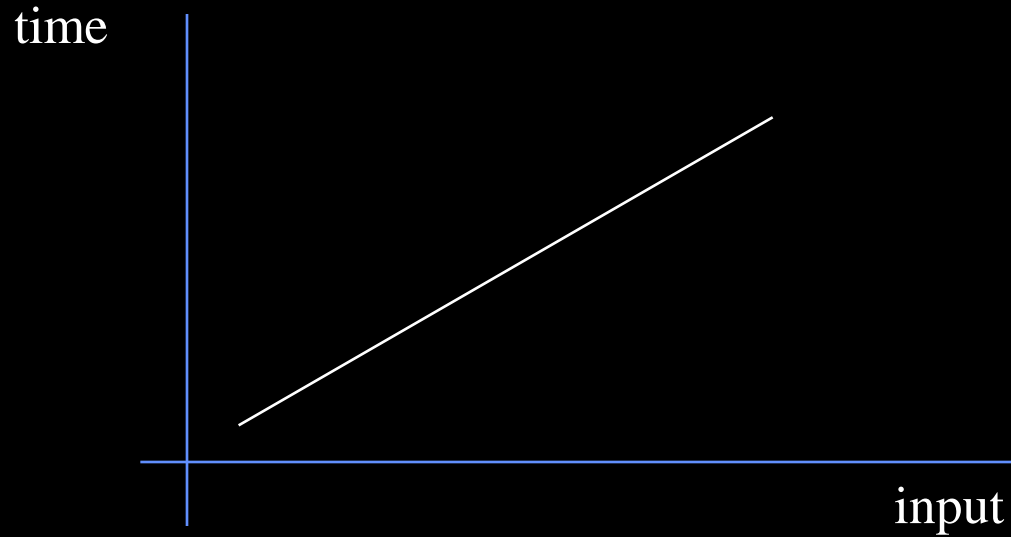
This notation is independent from any computer architecture.

## Complexity Function:

This notation defines the impact of input size on  
Time and Space parameters of algorithm.


# Complexity Function :

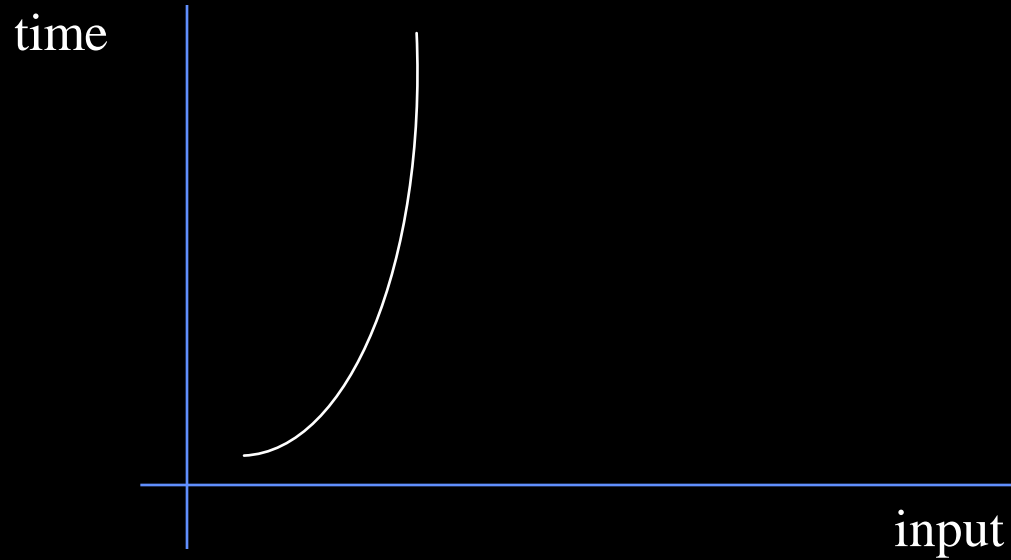
$O(n^t)$   polynomial





# Complexity Function :

$O(t^{f(n)})$   exponential



# Complexity Function :

Class	Complexity	Operation Count	Time
Constant	$O(1)$	1	1 micro sec.
Linear	$O(n)$	$10^6$	1 sec.
Quadratic	$O(n^2)$	$10^{12}$	11.6 day
Cubic	$O(n^3)$	$10^{18}$	32.000 year
Exp	$O(2^n)$	$10^{301.030}$	$10^{301.006}$

>> age of universe..

# Big Numbers :

Age of Earth	$10^9$ year
Age of Universe	$10^{10}$ year
Total life time of Universe	$10^{11}$ year
Total count of atoms in planet	$10^{51}$
Total count of atoms in Sun	$10^{57}$
Total count of atoms in Galaxy	$10^{67}$
Total count of atoms in Universe	$10^{77}$

# Complexity of a problem:

## Problem Classes

**Solvable:** these problems are solvable by a polynomial time algorithm.

**Unsolvable:** they are not solvable by polynomial time algorithm. **HARD !**

## Complexity Classes :

Categorical classifications of known problems according to their difficulty degree...

## Complexity Classes :

### Class P (Polynomial) :

If a problem can be solved in polynomial time by a deterministic computer, this problem is belong to

P

Class.

## Complexity Classes :

Class NP ( Non deterministic but Polynomial) :

If a problem can be solved in polynomial time by a non-deterministic computer (such as unlimited parallel computing capability), this problem is belong to

NP

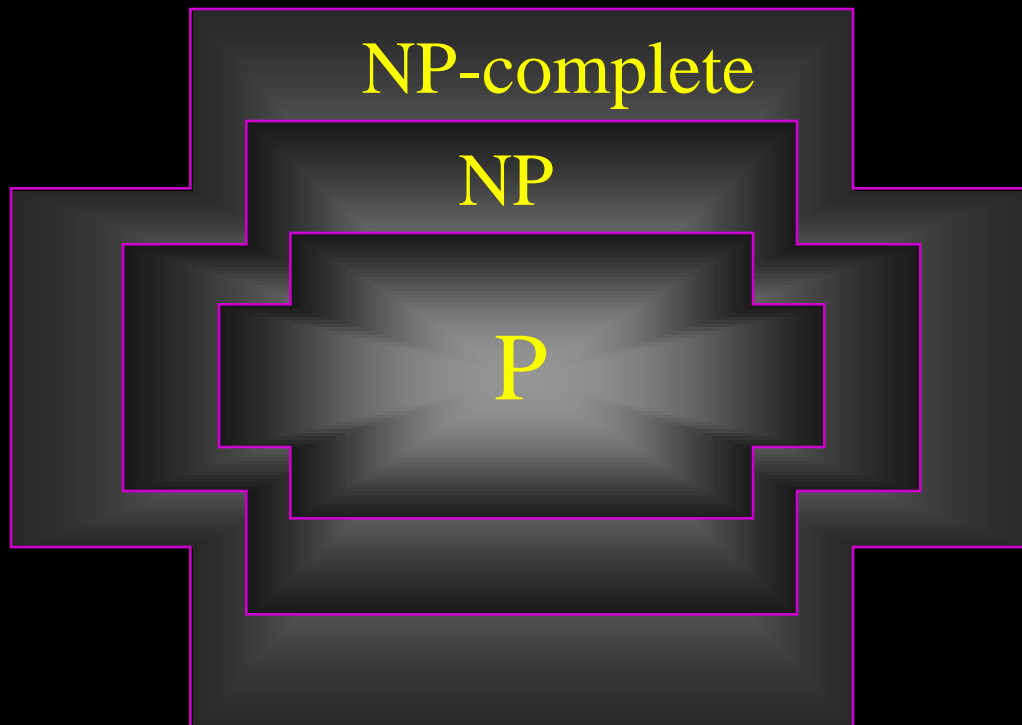
class.

## Complexity Classes :

**Class NP – Complete:** finding a solution for an instance of the NP-complete problem, which can solve other all similar problems of this NP class.



# Complexity Classes :



## Complexity Classes : Examples

Travelling Salesman : A salesman have to visit  $n$  different cities.

What is the shortest path which is every city visited only one time?

# Complexity Classes : Examples

Travelling Salesman :  $n=2, 3, 4$  and  $5$  is easy problem but if  $n=25$  or more !!??...

# Complexity Classes : Examples

**Factorization** : Finding of multiplicative prime factors of a number.

Example:  $60 = 2 \times 2 \times 3 \times 5$

# Complexity Classes : Examples-Factorization

Prime	Prime	Multiplication	Time
=====	=====	=====	=====
p	q	$n=p \times q$	
223	293	65339	30 sec.

**Class : P**

# Complexity Classes : Examples-Factorization

Multiplication	Prime	Prime	Time
=====	=====	=====	=====
$n=p \times q$	$p$	$q$	
65339	223	293	1 hour

**Class : NP**

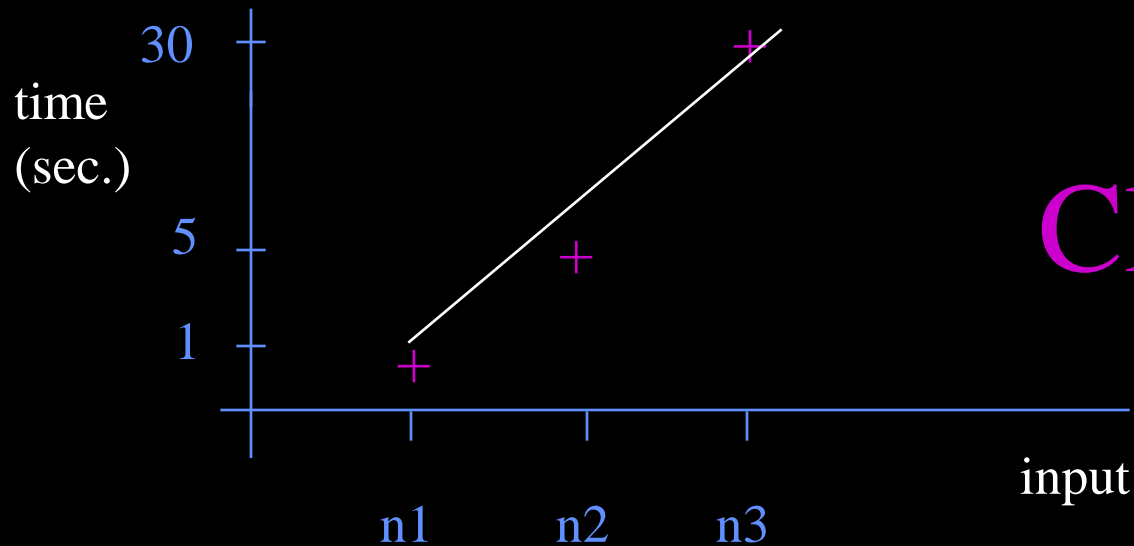
# Complexity Classes : Examples-Factorization

Example :  $n=p \times q$

input	operation	time
=====	=====	=====
$n1=p1 \times q1$	$3 \times 5 = 15$	1 sec.
$n2=p2 \times q2$	$11 \times 17 = 187$	5 sec.
$n3=p3 \times q3$	$223 \times 293 = 65339$	30 sec.

# Complexity Classes : Examples-Factorization

Example:  $n=p \times q$



Class : P



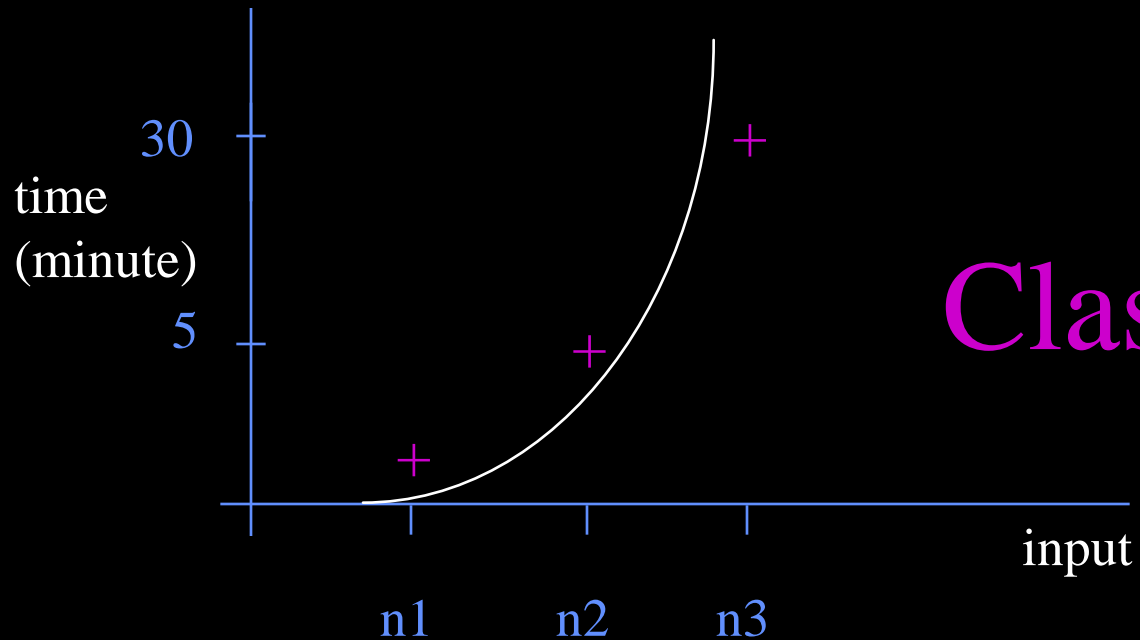
# Complexity Classes : Examples-Factorization

Example :  $n$  is the multiplication of two prime numbers.  
Try to find these prime numbers.

input	operation	time
<u>=====</u> n1	<u>=====</u> $15 = 3 \times 5$	<u>=====</u> 1 sec.
n2	$187 = 11 \times 17$	5 minute
n3	$65339 = 223 \times 293$	60 minute

# Complexity Classes : Examples-Factorization

example:  $n; p, q$



Class: NP

