

Tuesday.  
March 10.

Thursday - Quiz (mini-test)

Friday (3/13) - TCM

Tuesday 3/24 → project report

Thursday 3/26 → Longpre will teach complexity.

Problem: we need to separate feasible algorithms from non-feasible ones.

Examples:  $t \sim n^2, n^3$  → feasible

$t \sim 2^n$  → not feasible

Definition:  $t_u^w(n) = \max t_u(x)$

$x: \text{len}(x) = n$

An algorithm  $u$  is called feasible if there exists a polynomial

$P(n)$  such that  $\forall n: t_u^w(n) \leq P(n)$

Imp. fact: This definition doesn't fully reflect the notion of feasibility.

Example:  $t_u^w(n) = 10^{300} \cdot n$

Common sense viewpoint: not feasible

From our def: it is feasible (because  $10^{300} \cdot n$  is polynomial)

$t_u^w(n) = 10^3 \cdot \exp(0.000001 \cdot n)$

Common sense: feasible [becoz  $e$  is small]

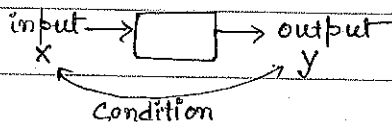
our def: not feasible [becoz  $\exp$  is not polynomial]

So, the definition has limits. No one knows how to produce a better definition.

We can use the knowingly imperfect definition of feasibility.

What is a problem?

In mathematics:



$X \rightarrow$  statement

$Y \rightarrow$  proof of  $X$  or  $\neg X$ .

comments: 1. checking whether  $y$  is a correct proof  $x$  is relatively easy.

2. The length of the proof should be feasible.

[checking the steps followed in the proof]

[like 4 color problem - length is too much]

condition  $R(x, y)$  is feasible means,  $\text{len}(y) \leq P_2(\text{len}(x))$  bounded by polynomial of  $(\text{len}(x))$

we have:-

\* feasible predicate relation  $R(x, y)$  true or false.

\* polynomial  $P_2$ : is the step by step start of a proof.

Given:-  $x$  binary seq.

we want:- find  $y$  such that  $P(x, y)$  and  $\text{len}(y) \leq P_2(\text{len}(x))$ .

In Physics:-

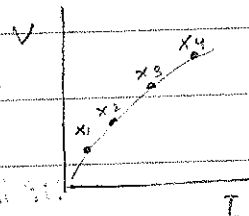
$x$  - observation

$y$  - theory: that explains observations.

$R(x, y)$  - check whether observations are consistent with the data.

here,  $\text{len}(y) \leq \text{len}(x)$   
 $\downarrow$  no of observations

The theory



$y$ :  $V = R \cdot I$

Engineers:-

$x$  - specifications (cost, how much it weight, it tolerate, wind etc.)

$x$ :  $x_1, x_2, \dots, x_n$   
no of observations

$y$  - spec' design [like constructing bridge]

$R(x, y)$  - feasible [now-a-days computer simulations help in testing the design like simulating a truck going over bridge]

but coming up with a good design is difficult.

$\text{len}(y)$  is not so complex here.

Humanity:-  $x$  - emotions

$y$  - a poem

$R(x, y)$  - not formalizable.

length,  $n$   
 how many strings with  $n \rightarrow 2$   
 with 1 length,  $2^1$  [0  $\rightarrow$  0]  
 2 " ,  $2^2$  [0, 1  $\rightarrow$  00, 01, 10, 11]

In principle, every practical problem can be algorithmically solved by trying all binary strings  $y$  of length  $\leq P_2(\text{len}(x))$ .

Problem: we need  $2^n$  time not feasible.

Good news:- If we guess an answer that we can check it's correctness in polynomial time.

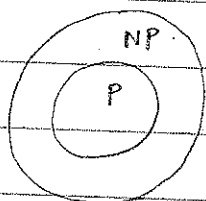
A P.R. problem  $\equiv$  can be solved on a non-deterministic TM in practical polynomial time.

Non-deterministic Polynomial = NP  
 NP  $\equiv$  class of all practical problems.

CS problem:—  $X$ —original list  
 $Y$ —sorted list.

$R(X, Y) \div$  checking if  $y$  is same list as  $X$  &  $n \in Y$  is sorted.  
, is easy  
 $\text{len}(Y) = \text{len}(X)$ .

P—class of all problems which can be solved by a feasible algorithm.  
example— sorting problem.



Natural question:—  $NP \stackrel{?}{=} P$

Is it possible to solve any practical problem in reasonable time? Ans:— we don't know.

NP hard problem:— Hardest problems in NP.

By, NP hard we can reduce class of NP to a smaller set of problems.