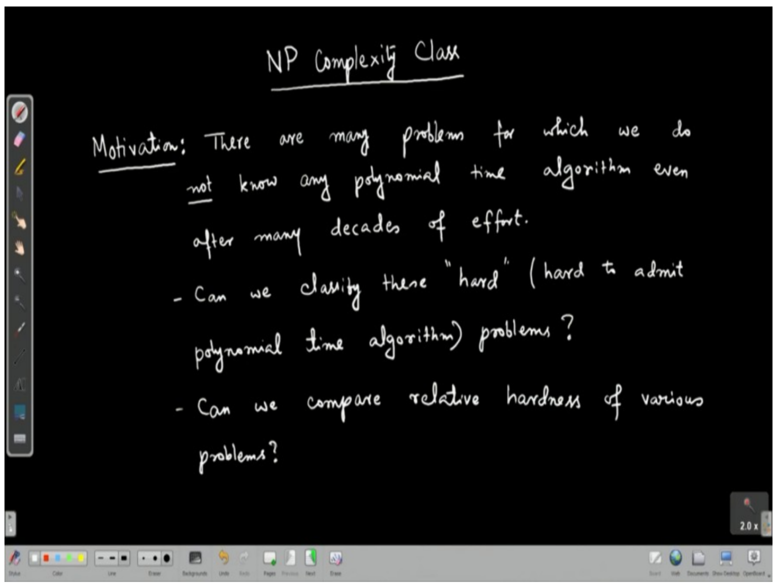
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Introduction of NP, Co-NP and P

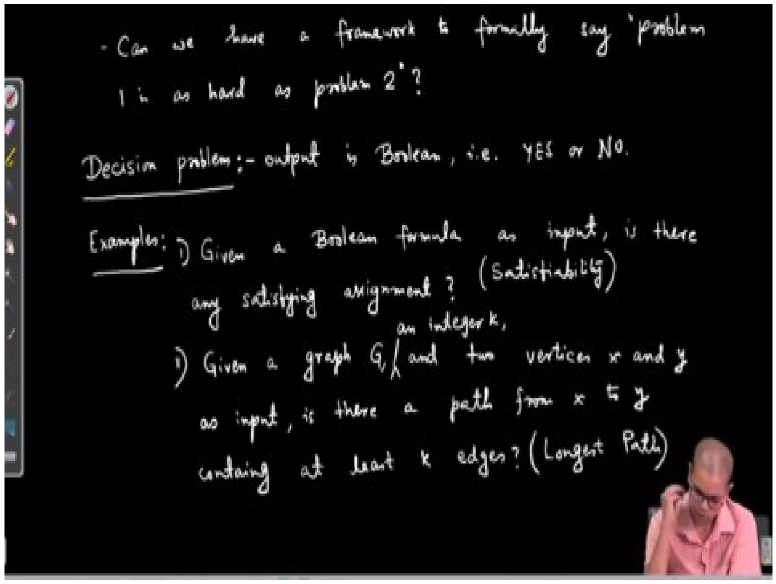
Welcome in the last class we have concluded randomized algorithms, so from today from this class you will study the framework of NP completeness formalizing when the problem is hard and when we do not expect to have a polynomial team algorithm and so on.

(Refer Slide Time: 00:49)



So, a new topic is NP completeness NP complexity class. So, what is the motivating question? There are many problems for which we do not know any polynomial time algorithm, even after many decades of effort. So, can we classify these hard means hard to admit polynomial time algorithm, can you classify these hard problems? Can we compare relative hardness of various problems?

(Refer Slide Time: 04:31)

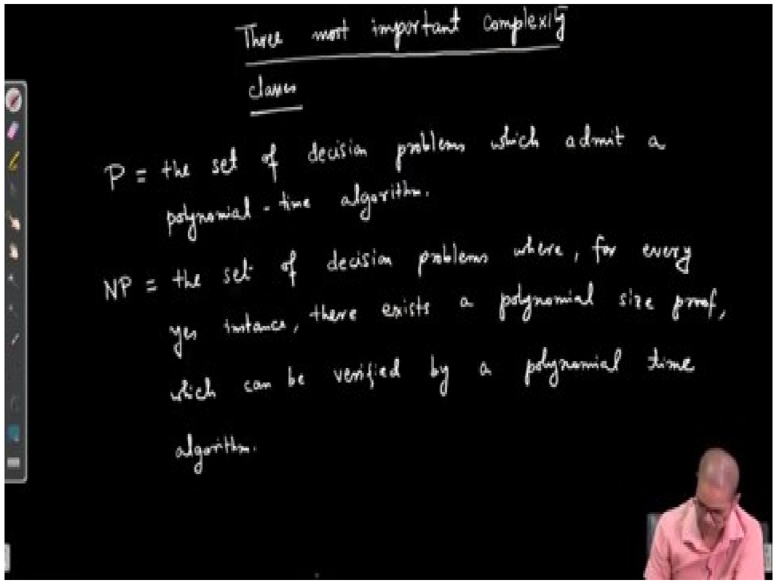


Since can we have a framework to say let me explain to formally say the problem 1 is as hard as problem 2? So, with these motivating questions let me present the framework of NP completeness, NP hardness and the complexity theory around it. So, for this and the some following lectures we will work on decision problems. So, what is the decision problem? A decision problem is a problem where output is Boolean that is YES or NO.

So, examples of decision problems say given Boolean formula has input is there any satisfying assignment. So, for two set problem we have seen an algorithm randomized algorithm and in general given a Boolean formula is there an assignment to the variables of the formula which makes its true and here we are only interested in yes, no version that means either say no or say yes.

Second example given a graph G, as input and two vertices as input is there a path from x to y containing at least k edges. So, graph G and integer k they are all input and two vertices x and y does there exist a path between from x to y containing at least k edges. So, here also we are not in interested in finding a path if there exists a path of containing at least k edges but and the interest is only finding the yes, no answer. So, these are decision problems.

(Refer Slide Time: 09:57)

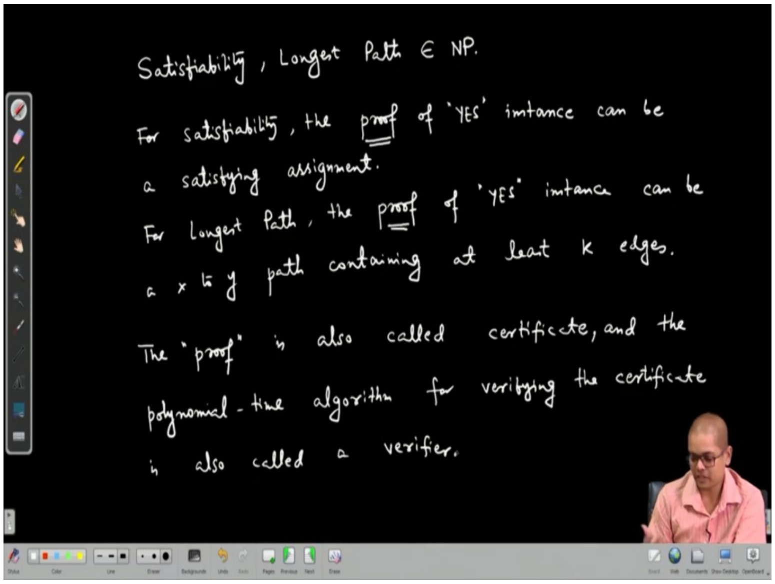


Now we classify the problems mainly into three classes, three complexity classes that we will see in this lecture of course there are many complexity classes which classify these problems and this itself is an interesting topic in computer science and area of advanced research. So, we will deal with three complexity classes. So, let me write three most important complexity classes which are most commonly used or is most commonly applicable.

First is P this is the set of decision problems which admit polynomial time algorithm. Then our next class is NP the set of decision problems where for every YES instance there exists a polynomial size proof which can be verified by polynomial time algorithm. For example, let us call the first problem satisfiability and the second problem longest path. So, for both problems you see that if the instance is the YES instance, then there exists a polynomial size proof.

Which is called certificate which can be verified by another polynomial team algorithm which convinces that the instance is indeed a YES instance. For example, for satisfiability the proof could be a satisfying assignment that means a true false assignment to the n variables and for this longest path problem the proof could be a path which contains at least k edges.

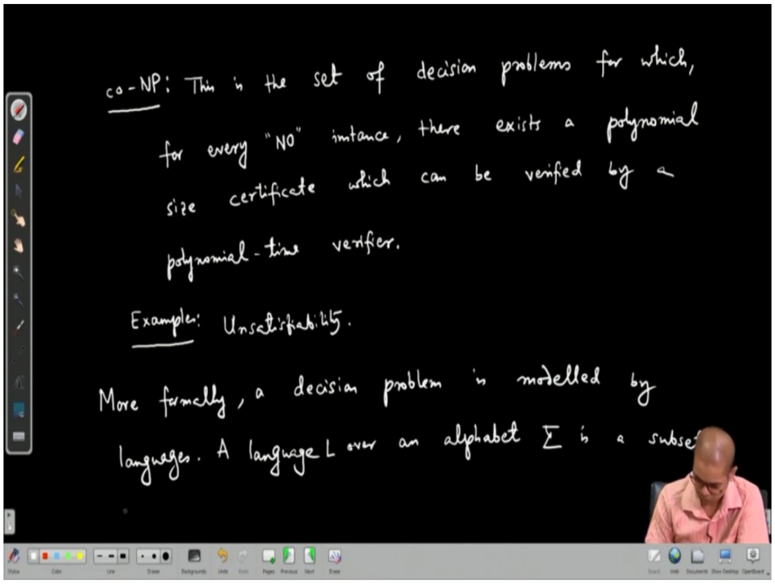
(Refer Slide Time: 15:04)



So, satisfiability then longest path belongs to this complexity class NP for satisfiability, the proof of YES instance can be satisfying assignment for longest path the proof of YES instance can be x to y path containing at least k edges. And these are the two problems among many problems for which we do not know any polynomial team algorithm even after decades of long effort and we will see soon that you know these problems are interconnected.

In the sense that if we are able to find a polynomial team algorithm for one problem then we will be able to find a polynomial for another problem. So, this proof is also called certificate and the algorithm and polynomial time algorithm for verifying the certificate is also called the verifier. So, this is the complexity class NP.

(Refer Slide Time: 19:26)

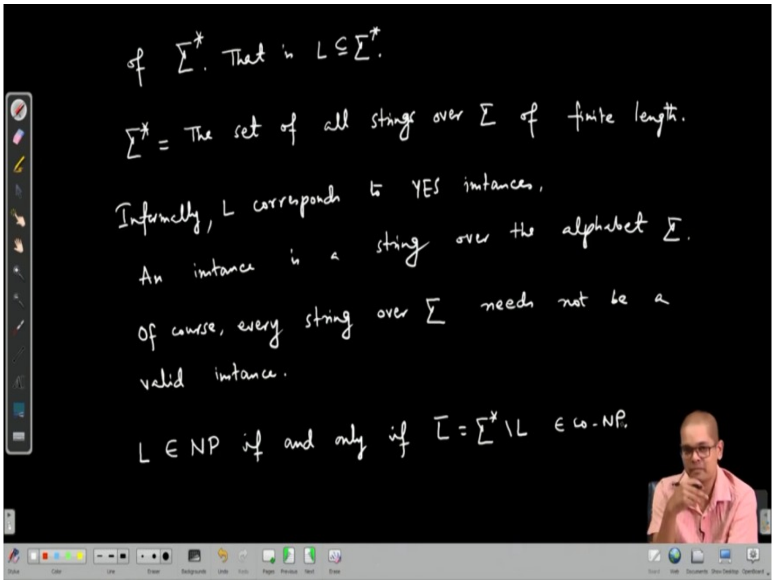


Our next complexity class is co NP which is like complement of NP is this class contains the problems for which the NO instances have certificate small certificate but Yes instance we do not know. So, this is the set of decision problems for which, for every NO instance there exists polynomial size certificate which can be verified by a polynomial time verifier. So, examples you know unsatisfiability given a Boolean formula is it unsatisfiable.

If it is yes that means it is unsatisfiable then you can give a satisfying assignment as a certificate of NO of the instance, so unsatisfiability. So, it is that is why it is co NP you take any problem in co NP any problem in NP and define the complement problem. Now what do you mean by complement problem? So, to do this formally, let us formalize the notion of decision problems more.

So, more formally a decision problem is modelled by languages. A language over and alphabet Σ is a language L is a subset of Σ^*.

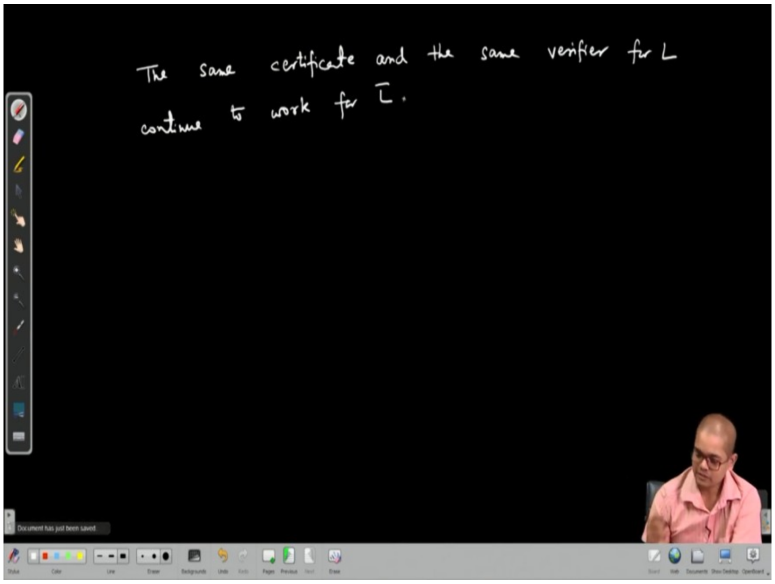
(Refer Slide Time: 23:52)



What is sigma star? That is L is a subset of Σ^*. Σ^* is the set of all strings over sigma of finite length. So, each language you know for satisfiability you can think of the corresponding language L instances. So, you know informally L corresponds to YES instances. And an instance is a string over the alphabet Σ but of course all strings should not be a valid instance every string over Σ needs not be a valid instance.

So, L basically corresponds to those strings which are valid instances and their YES instances. So, where does, what is the relationship between P, NP and co-NP? Of course, if L is in NP if and only if $\bar{L}$ which is $\Sigma^* \setminus L$ is in co NP, the same verifier and same certificate continues to work.

(Refer Slide Time: 27:39)



The same certificate and the same verifier for L continue to work for $\bar{L}$ because the same certificate of verifier was proving that there are some instances, YES instance. Now in $\bar{L}$, L bar the NO instances of $\bar{L}$, are precisely the YES instances. So, YES instances of L, so the same verifier and same certificate works for $\bar{L}$ also. That proves that you know L is in NP if and only if $\bar{L}$ is in Co-NP. So, let us stop here so, we will continue our investigation and study in the next class.