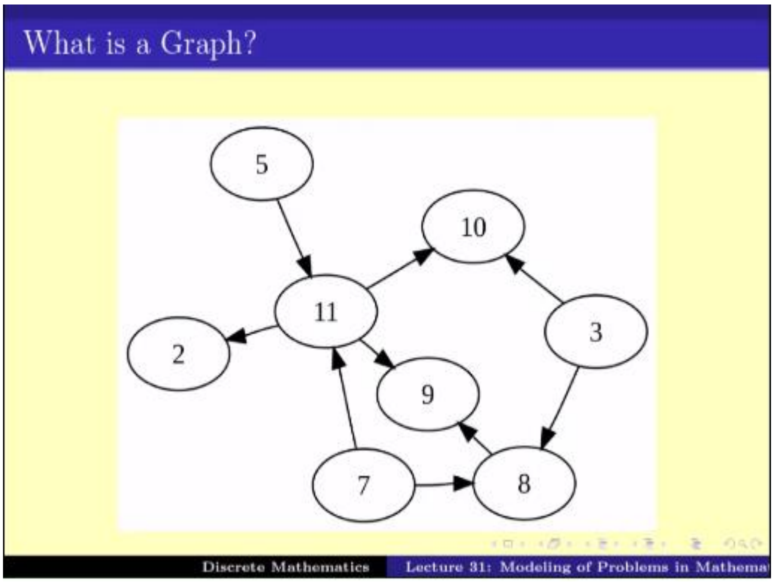
Discrete Mathematics
Prof. Sourav Chakraborty
Department of Mathematics
Indian Institute of Technology – Madras

Lecture - 31
Modeling of Problem (Problem 3 and 4)

Welcome back. So we have been looking at how to model the problems using Linear Programming and Graph theory. The basic idea is that if we have to solve the problem or if we have to attack the problem one of the easy ways is to convert it in a Mathematical language that is well studied. In that ways you can use the known Mathematical tools or use the Mathematical tools to solve the problem.

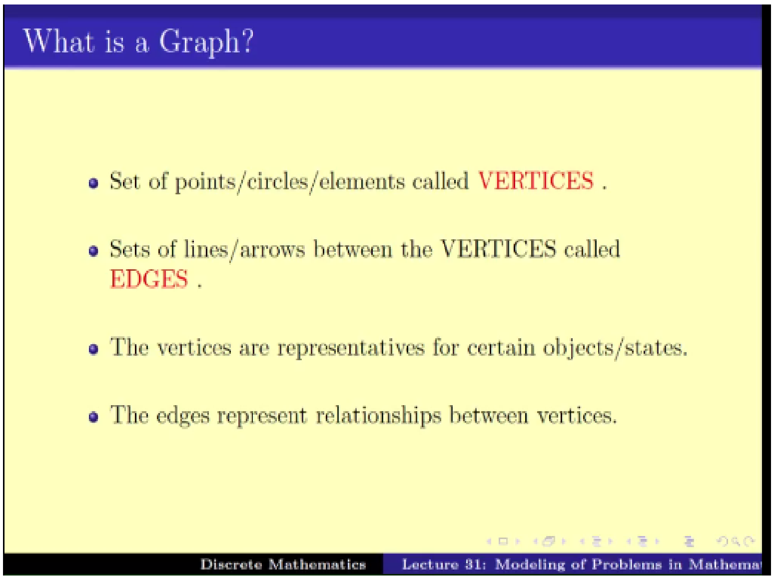
There are various Mathematical languages which can be used to model a problem. For discrete problems two of the more powerful techniques are Graph theory and Linear Programming. So in the last couple of videos we have looked at some problems and how to solve them using Linear Programming. Now in this video, we are back to Graph theory and we will see how one can use it to solve the problem using Graph theory.

(Refer Slide Time: 01:14)



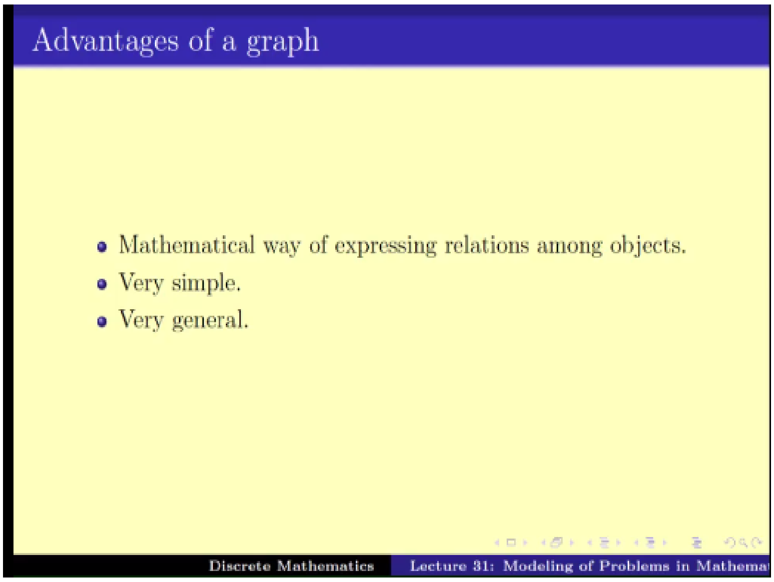
So to quickly revise what a graph is? A graph is something like this.

(Refer Slide Time: 01:24)



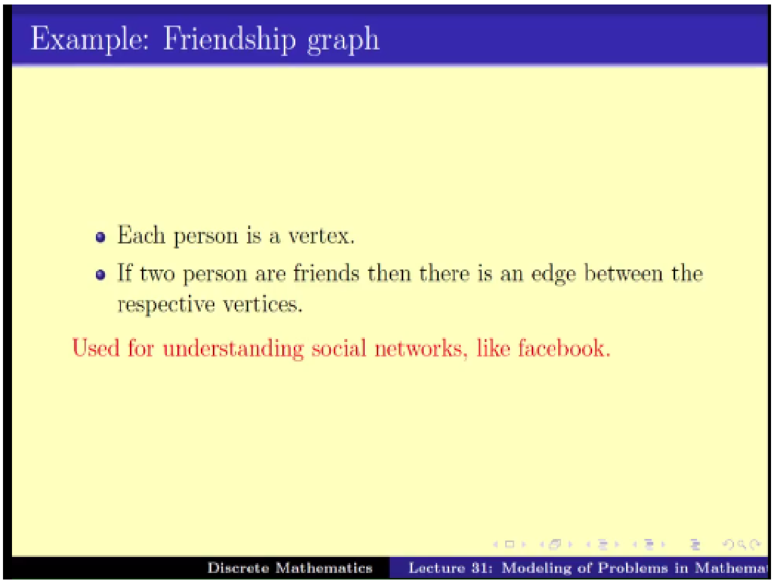
We have a set of vertices and the set of edges which are basically arrows or lines between the set of vertices. The vertices represent certain objects or states or something like that. The edges represent relationship between the vertices so binary relations in particular and using this kind of a structure we can represent various problems. So, for example, this is a set of vertices and this is a set of edges that we have.

(Refer Slide Time: 02:03)



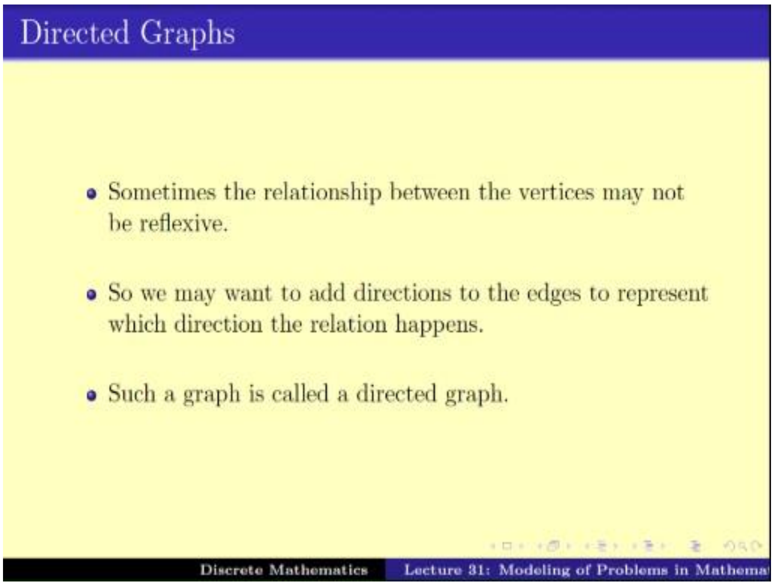
Of course, advantage of a graph is that is it is simple and is very general. It is used for representing expressing binary relations among objects. We have seen some of the examples already.

(Refer Slide Time: 02:17)



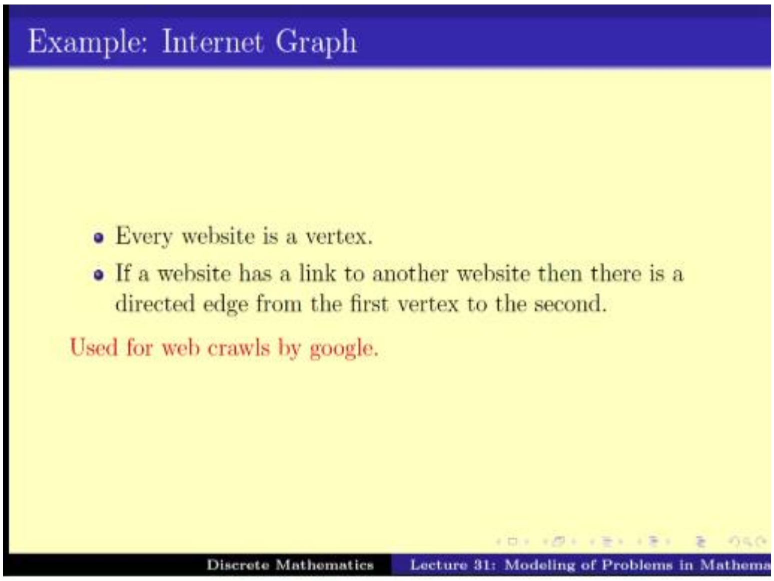
A typical example is a friendship graph where every person is a vertex and if two persons are frames and there is an edge between the respective vertices and we get a friendship graph using this technique.

(Refer Slide Time: 02:35)



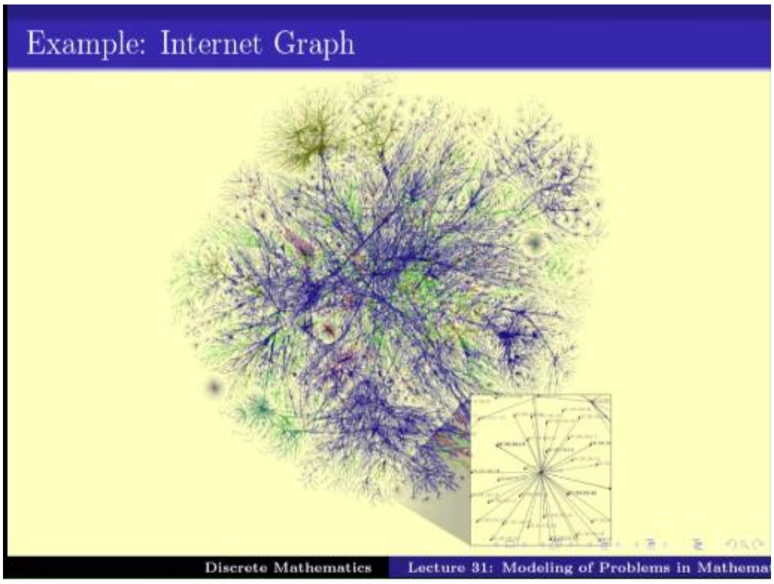
Or we can have directed when the relationship is not reflecting. In that case, we can add edges in one direction I mean the edges can have direction and we call directed graphs. So the edges can have directions and we have what we call that directed graphs.

(Refer Slide Time: 03:05)



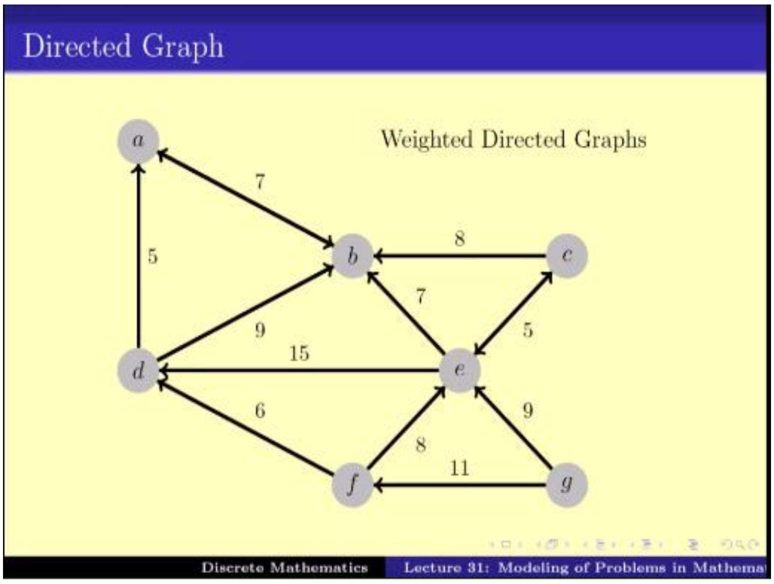
Now typical example of a directed graph is the internet graph where every website is a vertex. And if there is a link from a website to another website it is a directed graph and we get the internet graph which is used a lot by various search engines.

(Refer Slide Time: 03:24)



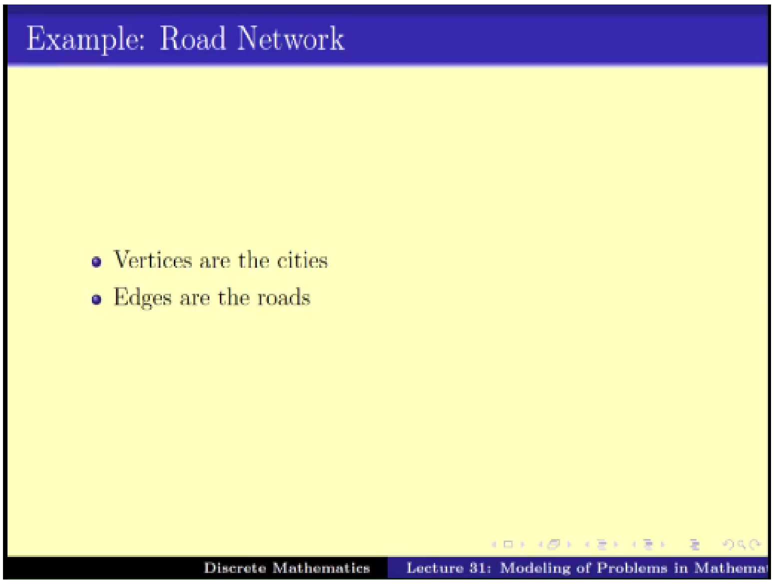
So this is for particular kind of directed graphs. Do you think it is a shot of a internet graphs. We have also have weights on the edges and in that case we have weighted directed graph.

(Refer Slide Time: 03:43)



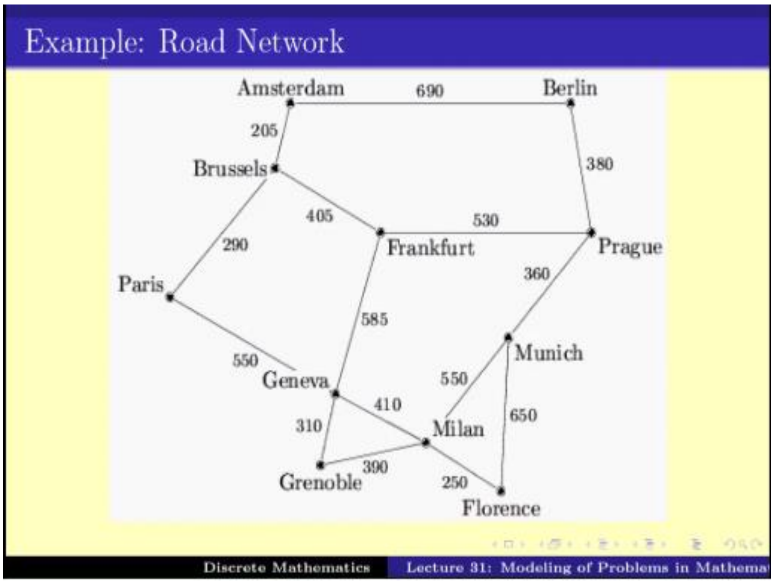
For example, we can have weights like this and we get a weighted directed graph.

(Refer Slide Time: 03:52)



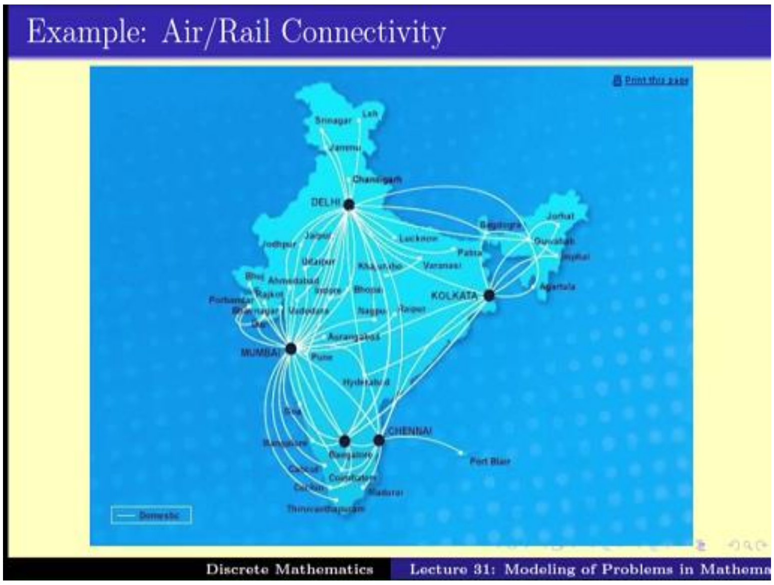
For example, because of road networks we can have vertices that are cities edges that are roads and the weights can represent the distance between two vertices.

(Refer Slide Time: 04:02)



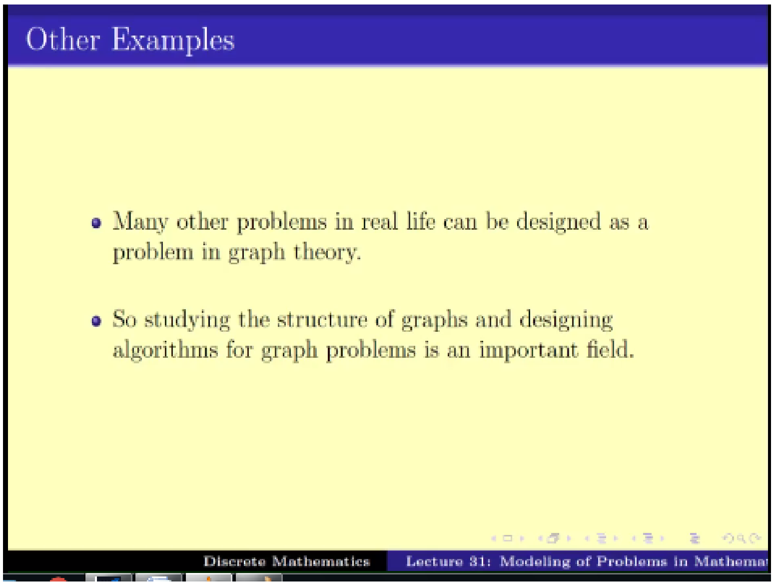
For example, distance between Amsterdam to Berlin is 690 miles and so on. Similarly, air connectivity, we can have if there is a vertices or airports and whether this is an edge between them one vertex to the another if there is a flight between those two vertices.

(Refer Slide Time: 04:28)



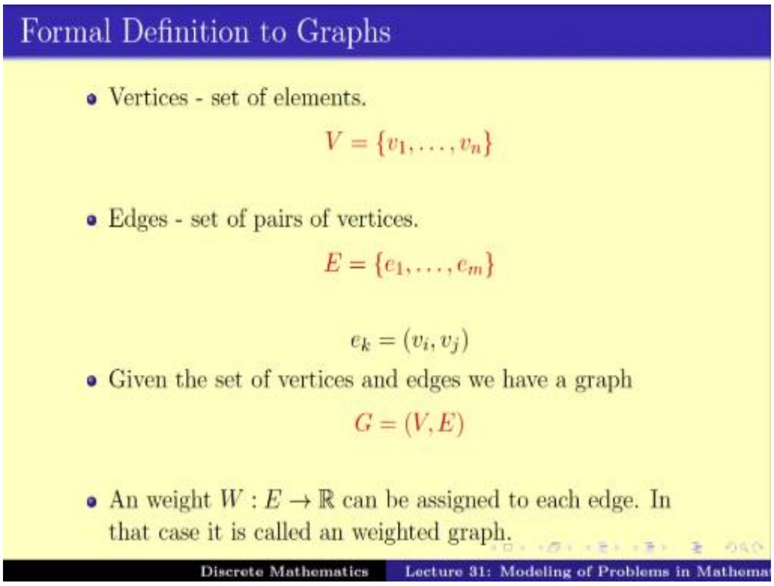
Typical example you can see it in almost any airplane this map is usually there it is a directed map of the air connectivity.

(Refer Slide 04:45)



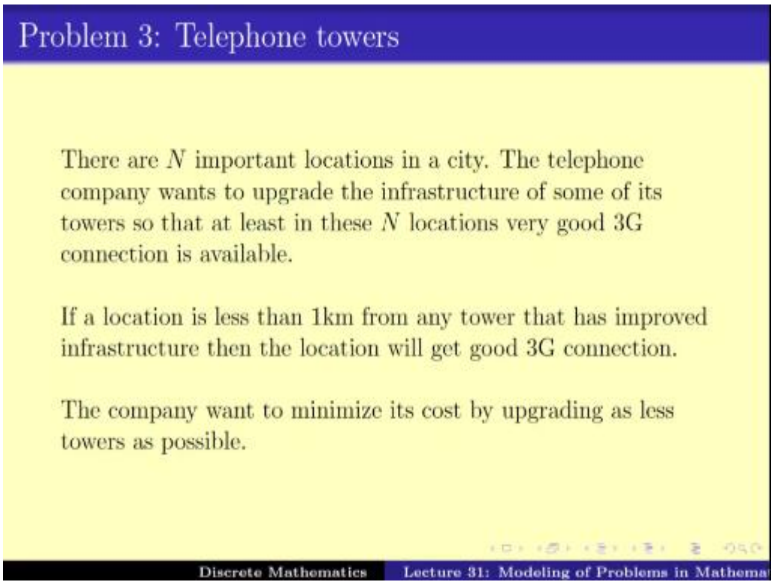
There are many other problems that can be converted into Graph theory and hence studying structure of graph and designing algorithms for graphs is an very important field.

(Refer Slide Time: 04:59)



A formal definition we have seen this thing a many times we can have vertices which is set of elements, edges which are pairs of objects, pair of vertices. The graph is given as the vertex and the edge and there can be the edges can be weighted. Namely, there can be weights on every edge. So there is a weight function from the edge set to the real numbers.

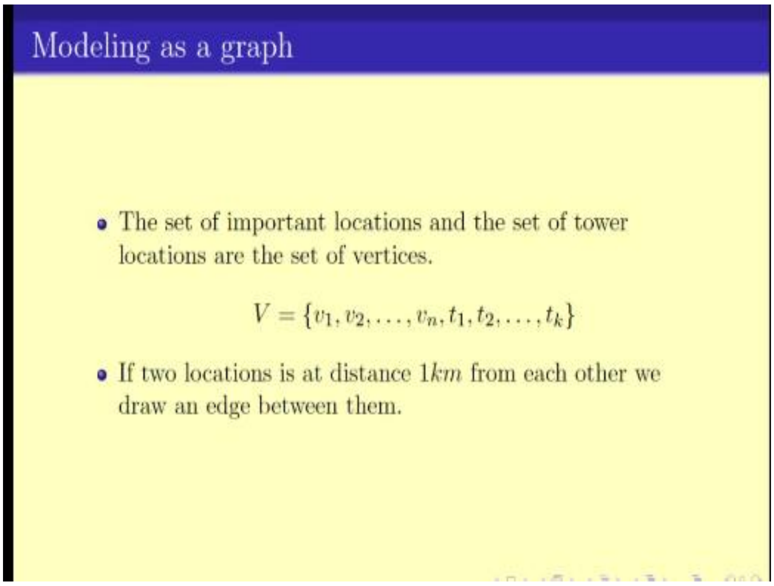
(Refer Slide Time: 05:27)



Now using all these things let us try to first solve the problem of the telephone tower problem that we spoke about last video. So here basically you have N important locations and we want to ensure that all the N important locations have 3G network. So for that to ensure that a location has 3G network we have to first upgrade the infrastructure of some of the towers and every of this N location must be in a distant less than 1 kilometer from at least one towers.

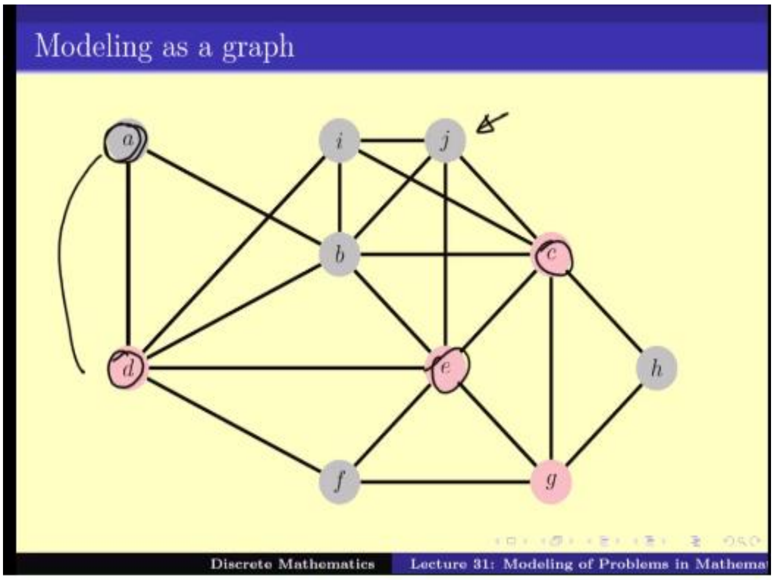
And it is still an optimization problem because we want to minimize the number of towers we want to optimize.

(Refer Slide Time: 06:26)



Now let us try to model it as graphs. So let the important locations hence the v1 to vn and the towers be in location T1 to TK. And if two of the locations are in distance 1 kilometer then we will draw an edge between them. So the vertex there is V1 to VN and T1 to TK and if there is a location if two locations are close to less than 1 kilometer distance we draw a edge between them.

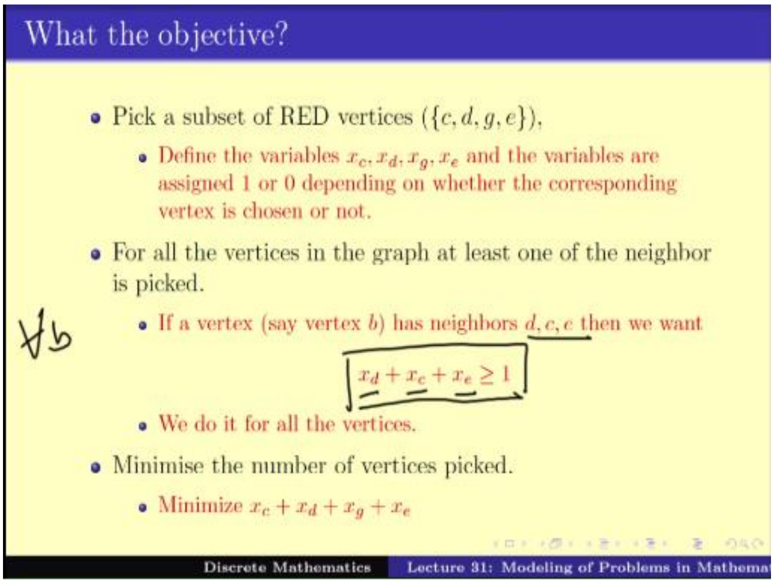
(Refer Slide Time: 07:04)



By doing so, we get a graph of this sort something like this where the red vertices are the tower locations and the gray vertices are the non towers locations and now what is it that we want. Now what does this graph say? It says that this vertex A is distance less than 1 kilometer from D, but this for the vertex J is distance 1 kilometer from both C and E. So if A has to get the 3G connection then D must be upgraded whereas if J has to get a 3G connection either C or E has to upgraded and so on.

Now clearly particularly this is a toy example in the sense that the graph is not an original graph, so it does not –exactly show all the complexities of this problem.

(Refer Slide Time: 08:08)



But the main idea is we have to pick a set of rare cases Red vertices such that for all vertices in the graph at least one neighbour is picked, one of the rare neighbour's is picked and we want to minimize the number of vertices picked. Now this is a typical problem we call it as a Vertex Cover Problem, but this is just a formulation of the problem how do you come and solves the minimum vertex of a problem is a different matter altogether.

So here I have used the graph tools, write it down another Vertex Cover Problem. Now how do you solve them? So first of all, one thing you do is that we can now use LP formulations on top of this. So, for example define for every vertex which is red let me defines the variables XC, XD, XG, XE representing whether I choose that vertex or not which is 1 if I choose the vertex 0 otherwise.

Now if I do it like that how do I ensure that all the vertex are at least one neighbour pick. So that means take a vertex at B and if it has neighbour C,B and E then we have to ensure that XC+XD+XG +XE > 1. This ensures that this is the XC, XD, XG, XE is 0 or 1. This can be greater than 1 only if one of them is 1 which means one of its neighbours is picked. So if we basically put the condition that for all vertex B withholds then we have the condition, the whole condition.

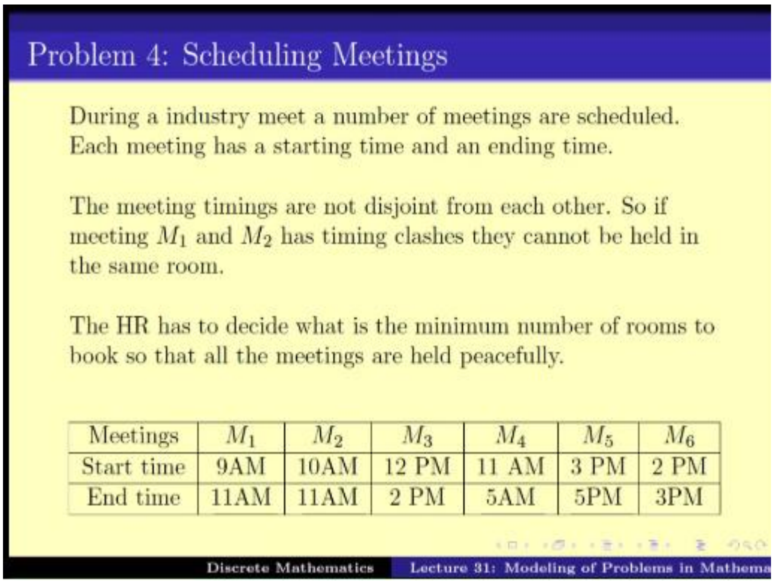
And we are trying to minimize something. The minimization is of course we have to minimize some work all the number of things that we are picking up not. It is not that here we picked up the problem converted into a graph problem and kind of helped us to visualize what is going on and that what helped us to write the LP formulation of this. Now this is a of course a Vertex Cover Problem and it is very well studied problem.

And hence there is a big literature on this particular problem also. But in any case now that we have done we have a set a variables. We want to minimize something such that this thing holds and the variables are 0 and 1 and this is exactly the LP formulation of this problem. Again as you might we call this is not exactly LP formulation because it has the variables is not from or but it is from 0 and 1.

And hence we do not know how to solve this particular problem exactly. So we have to solve this variable. Unfortunately, it is an integer Linear Programming and we might have to apply things like randomization or the rounding technique that we talked about last class to solve it. Of course, we can relapse the LP to get relapse this condition (()) (12:01) to the real number between 0 and 1 in that case we get a reality we can solve it.

And then we can use the usual technique that we did last time to solve it. Now the main point here is not how to solve it, but the fact that we can take the problem, use Graph theory to model it or visualize it and then use Linear Programming to model that problem into defined setups.

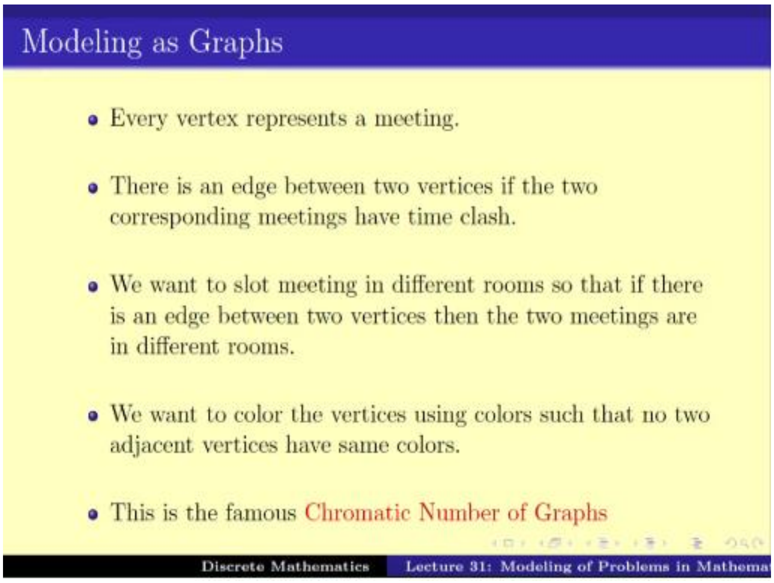
(Refer Slide Time: 12:36)



Now let moving on to an another problem. Here is the scheduling problem that we have. Here we have this 6 meetings that we have and the HR has to ensure what is the least number of room to book such that it can arrange for the meetings we held in different rooms so that there is no confusion or other two meetings which are clashing in time does not happen in this same room. What is the minimum number of room to meet that?

Now again we can use Graph theory to solve this or to model this problem.

(Refer Slide Time: 13:25)

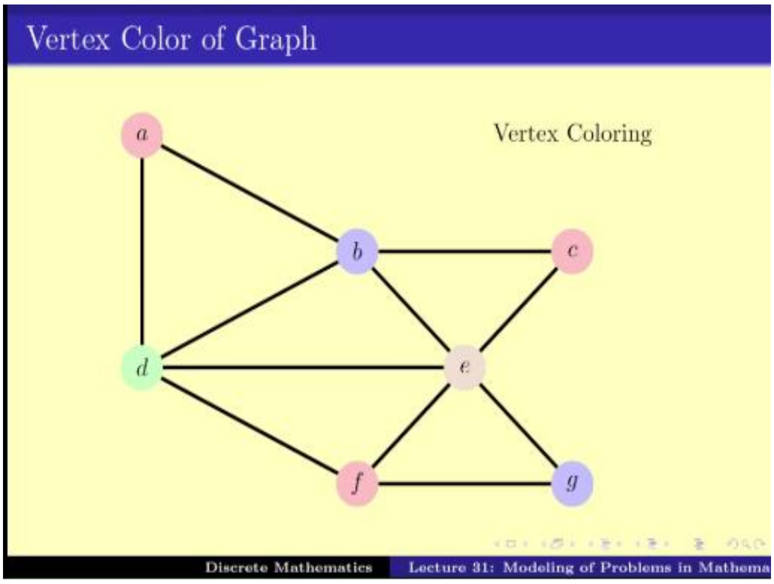


So here we represent every meeting as a vertex and we draw an edge between two vertices if there is a clash in time and we want to slot them in different room so that if there is an edge between two vertices they happen in two different rooms. So in other words, we want to colour the vertices using as little as less number of colours as possible. And this is what of course we have done this problem it is the famous Chromatic Number of Graphs.

So in other words, we want to kind of assign the vertices to rooms in other words colours such that no two adjacent vertex has the same colour. Question is that, what is the minimum number of colours required to colour the graphs?

(Refer Slide Time: 14:24)

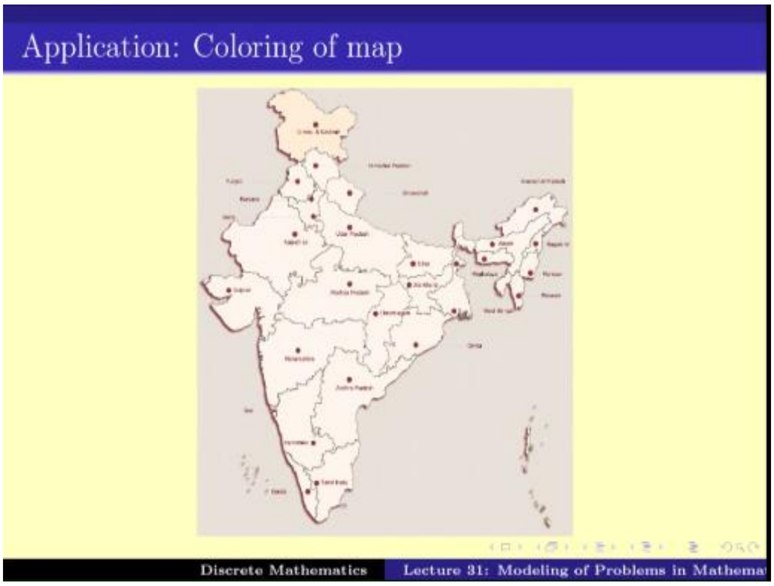
on.



So if this is the vertex of course if we colour if we assign this one to the Red room then B must be assigned to some other room which is the blue room and then C can be assigned to Red room again then B has to be assigned to a new room cannot be assigned to Red and Blue because A and B of Red and Blue and similarly you go on and we can assign various vertices to the various rooms.

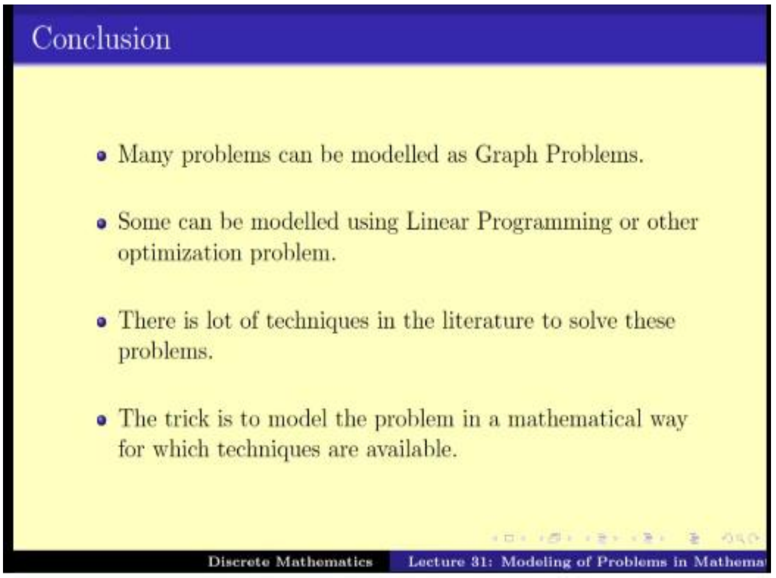
Question is that what is the minimum number of colours required to colour this graphs?

(Refer Slide Time: 14:58)



The application of course as I have pointed out is available in various coloring of map and so on.

(Refer Slide Time: 15:06)



Now the main idea that I wanted to convey in this few video lectures is that many problems can be modelled as graph problems. Some can be modelled using Linear Programming or optimization problem. There is lot of technique in literature to solve this problem. One important thing to notice how to model this problem in a Mathematical way, in different Mathematical language.

Once, you know, how to write it in a different language then you can use tools from other techniques to solve it. We have seen Graph theory and Linear Programming both of them are subject by itself of whole course subjects. So I cannot go too much into details of Graph theory and Linear Programming, but here the main point is that learning how to convert a problem into a Graph theory problem or a Linear Programming problem or in some other language is a very important and essential tool to have.

In the next video, we will look at other aspects of this map we will be moving away from Graph theory.