Advanced Operations Research

Prof. G. Srinivasan

Department of Management Studies

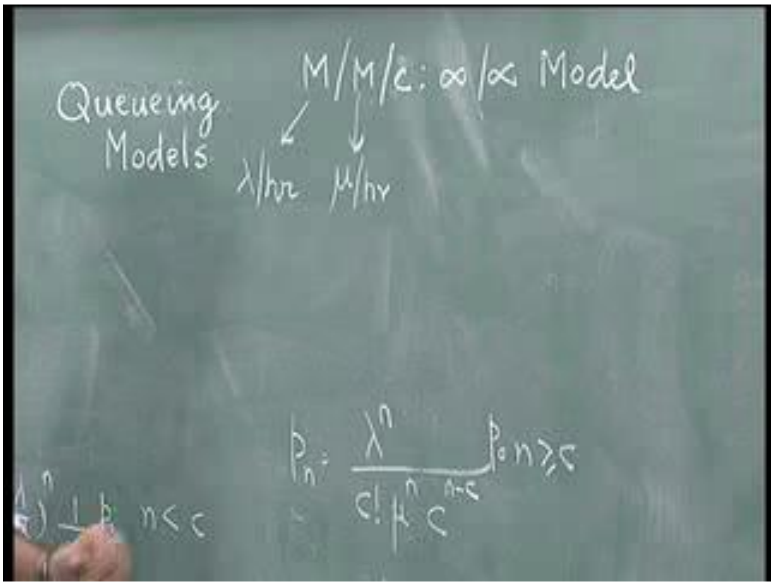
Indian Institute of Technology, Madras

Lecture - 32

Multiple Server Queueing Models

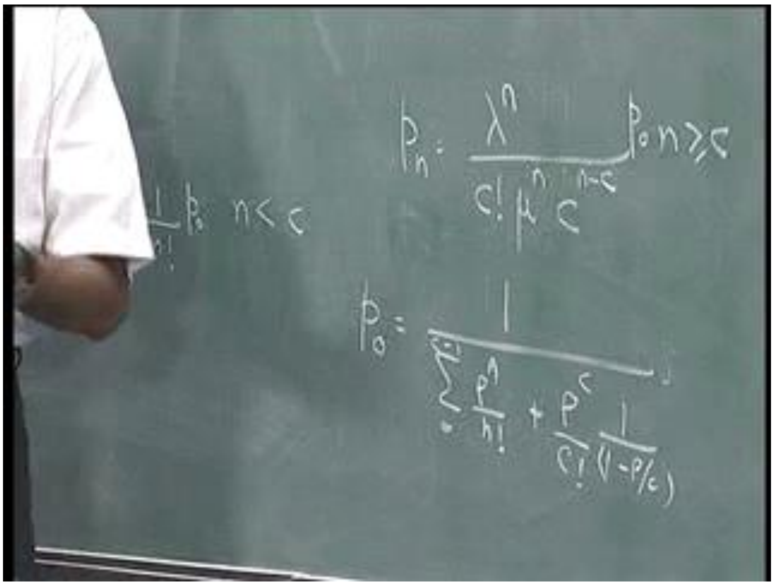
In this lecture, we continue our discussion on the queueing models and on M/M/c infinity infinity model where we have multiple servers and we have an infinite queue length.

(Refer Slide Time 00:40 min)



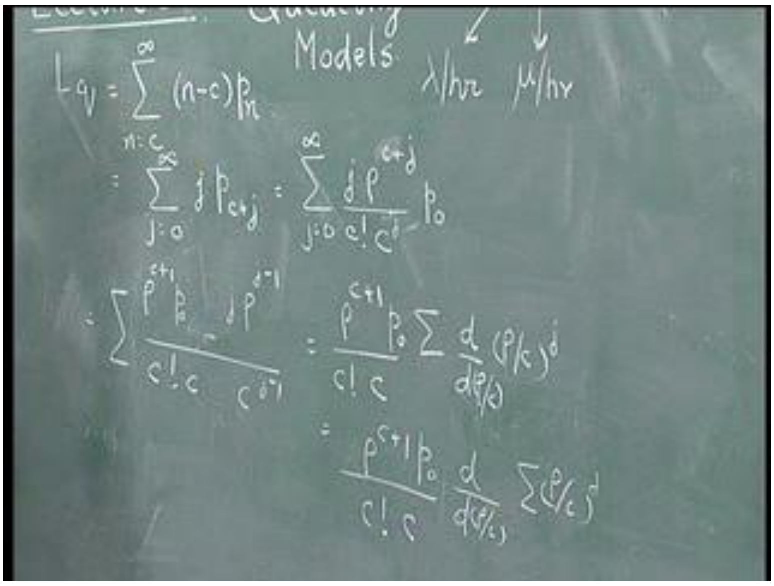
In the earlier lecture we derived some equations. We derived this expression for p_n probability that there are n people in the system as lambda by mu to the power n by n factorial p_0 when n is less than c and p_n is equal to lambda by mu to the power n divided by c factorial c to the power n minus c p_0 when n greater than or equal to c.

(Refer Slide Time 00:51 min)



This has infinite queue length, so p_0 plus p 1 plus p 2 up to p infinity equal to 1 gave us this equation p_0 is equal to 1 by sigma n equal to 0 to c minus 1 rho power n by n factorial plus rho power c by c factorial 1 by 1 minus rho by c. This was an infinite geometric series, this has as many terms depending on the number of servers so this will have 0 this will have exactly c terms from 0 to c minus 1.

(Refer Slide Time 02:05 min)



Now once we get an expression for p_0 we need to derive expressions for Ls Lq Ws and Wq. For multiple server models it is customary to define Lq first and get an expression for Lq from which we would get expressions for Ls Wq and Ws. So if we derive the equation for Lq we would get Lq is equal to sigma n minus c p_n, n equal to c to infinity. The queue is actually 0 when we have c people, the queue length is 1 when we have c plus 1 and so on, so n minus c into p_n would give us the corresponding expected length of the queue.

Now we need to substitute these expressions for p_n into this and we will get; so Lq is equal to this is sigma j equal to 0 to j equal to infinity j$p_{\text{c plus j}}$ so the queue actually starts when n is greater than or equal to c so we do not require this expression to get Lq and it is enough if we use this expression (Refer Slide Time: 3:10) to get Lq because here c to infinity has been substituted by 0 to infinity where j equal to n minus c. So j equal to 0 1 2 3 would mean n is equal to c, c plus 1 c plus 2 etc so it is enough to use this expression. So this will be equal to sigma j equal to 0 to infinity j lambda by mu to the power n rho power n by c factorial c to the power n minus c. So p probability that there is c plus j people in the system would give j rho to the power c plus j divided by c factorial into c to the power j p_0.

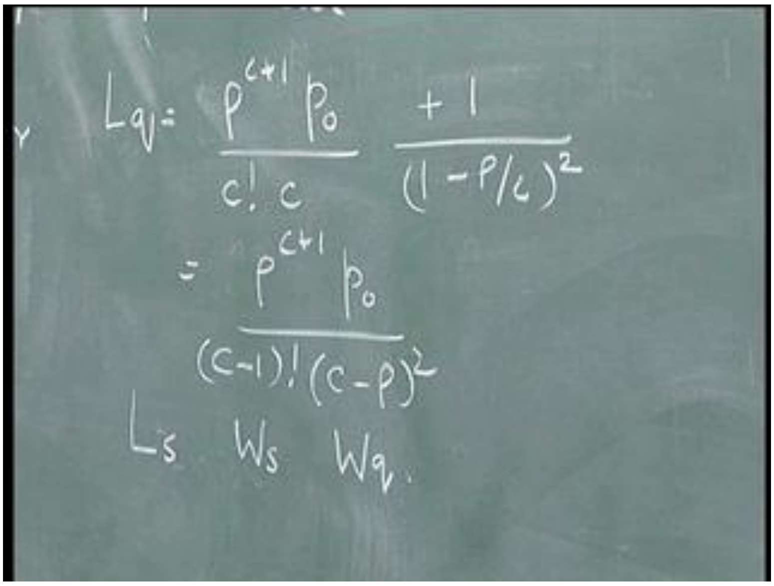
Now this comes because the general expression for p_n is rho to the power n by c factorial c to the power n minus c. Now n becomes c plus j so c factorial remains here so lambda by mu to the power c plus j is rho to the power c plus j. Now when n is equal to c plus j c plus j minus c would give only j so c factorial c power j p_0. Now from this we can write this as this (Refer Slide Time: 5:07) here we will pull out rho power c plus 1 such that we will get j rho to the power j minus 1 here divided by c factorial.

Here we have rho to the power j minus 1 so we need c to the power j minus 1 so we pull out another c here c factorial into c into p_0 by c to the power j minus 1 the summation actually comes outside here. This is the summation. Now his is rewritten as this (Refer Slide Time: 5:58) now we pull out some of the common things so rho to the power c plus 1 divided by c factorial into c comes out p_0 also comes out sigma we have j rho by c to the power j minus 1 which is d by d rho of rho by c to the power j.

Now d by d rho of rho by c to the power j would be j rho by c to the power j minus 1 into 1 by c so we will have a c factorial. Now d by d rho of rho by c, this is d of rho by c d by d rho by c of rho by c to the power j, so this will be j rho by c to the power j minus 1, so we have j rho by c to the power j minus 1. Now we remove the differentiation and the summation to the other side so we get rho to the power c plus 1 p_0 c factorial into c to d by d of rho by c sigma rho by c to the power j.

Now this is an infinite geometric series so this is rho to the power c plus 1 p_0 by c factorial into c d by d of rho by c. This will be 1 plus rho by c plus rho by c the whole square and so on. So it is an infinite geometric series with first term equal to 1 common factor being rho by c and we know that rho by c is less than 1. When we have an infinite queue length model we can use the infinite geometric series formula to get 1 by 1 minus rho by c.

(Refer Slide Time 08:09 min)

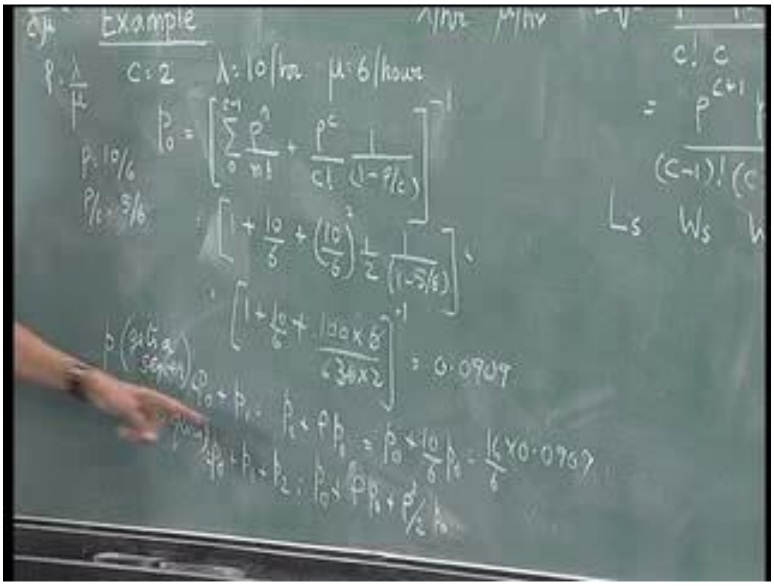


Now we can differentiate this with respect to rho by c to get Lq is equal to rho power c plus 1 p_0 by c factorial into c this will become minus 1 by 1 minus rho by c the whole square into minus 1, so we get plus 1 by 1 by 1 minus rho by c by differentiating with respect to rho by c, so this is like d by d x of 1 by 1 minus x. So 1 minus x will give minus 1 by x square and then you get another minus 1 that comes here so we will plus 1 minus rho by c the whole square. This on simplification would give us rho to the power c plus 1 p_0 by this is c minus rho the whole square we take a c square outside so one c will get cancelled here and one c will go here to get c minus 1 factorial into c minus rho the whole square into p_0. So this is the expression for Lq when we have an M M c infinity infinity model

And once we know Lq, then we can go back and derive expressions for Ls Ws and Wq using the earlier equations that we have derived for using the Littles formula. So once we know rho and c we get the expressions for p_0, which we have derived earlier from which we can get an expression for Lq and then using the Littles formula we can go back and get Ls Ws and Wq respectively.

Now we take a numerical example to try and understand how these values effectively look like.

(Refer Slide Time 10:30 min)



Now let us have a multifacility system, where we have c equal to 2 let us say we have typically like a web browsing center with two terminals. Let us say people arrive with lambda equal to 10 per hour and the effective service rates are such that we have mu equal to 6 per hour. Now this would give us lambda by c mu is 10 by 12, which is 5 by 6, which is less than 1 so lambda by c mu is less than 1.

Now let us find out the values. Now p_0 is equal to sigma rho power n by n factorial plus rho power c by c factorial into 1 by 1 minus rho by c, so this is what we derive as the expression for p_0 in inverse of this 1 by sigma rho power n by n factorial rho power c by c factorial into 1 minus rho by c. Now this would give us this sigma is from 0 to c minus 1. So when n equal to 0 we get or on expansion we would get this has two terms because c equal to 2 for n equal to 0 and for n equal to 1. So this would give us 1 plus rho power n by n factorial rho is lambda by mu so lambda by mu is 10 by 6. So these are the two terms corresponding to this, plus rho power c by c factorial, which is 10 by 6 the whole square into 1 by 2 c being 2 c factorial is 2 into 1 minus 1, 1 by 1 minus rho by c rho by c is 5 by 6. Now this term is 1 by 6 we can simplify this to get p_0 is equal to so this is 1 plus 10 by 6 plus this is 100 into here we have 1 minus 5 by 6 is 1 by 6. So this will go to the other side so into 6 by 36 into 2 inverse, so this will go 6 times so this on simplification would give us p_0 equal to 0.0909 that we have here.

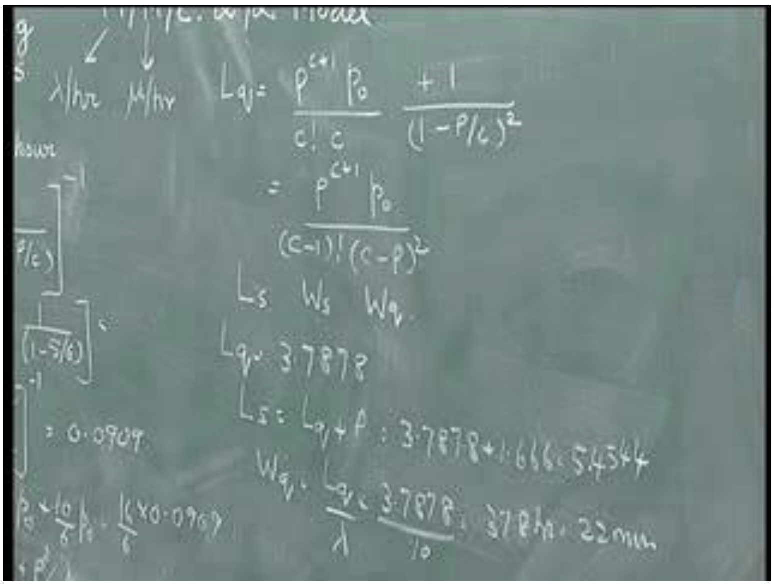
So p_0 is equal to 0.0909 and once we know p_0 we can calculate p 1 p 2 or any other number that we would like to do and once we know p_0 we can actually find out the rest of them.

If we want to know that the first question that we may look at is, what is the probability, that both the computers are free when a person arrives which means when a person arrives the entire system is empty so p_0 is when there is nobody in the system. So both the computers are free when the person arrives is given by 0.0909. The next one is what is the probability, that a person who comes and gets a computer.

Now there are two computers so a person who comes will get a computer, when both are free or when one is free, which means that will be equal to p_0 plus p 1. In this case it will be p_0 plus rho p_0, because we have n equal to 1 so we will simply have rho power n by n factorial so rho power 1 by 1 factorial into p_0, so that will become p_0 plus rho p_0. Rho in this example is rho is 10 by 6 and rho by c is 5 by 6, which is less than 1. So this probability will be this p_0 plus 10 by 6 p_0, which are 16 by 6 into 0.0909.

Another question would be what is the probability that there is no queue? So there will be no queue, when either there are zero people in the system or one person in the system or two people in the system. So the probability that there is no queue given by p_0 plus p 1 plus p 2, so this is probability that person gets a computer or gets a server this is probability of no queue. So this will become p_0 plus p 1 or p 1 is rho $p_{0,}$ so this will become rho p_0 and this will become rho square by 2 into $p_{0,}$ because p2 is rho square by 2 factorial which is rho square by 2 into $p_{0.}$ So we can substitute for p_0 and rho and rho square by 2 to get such a probability.

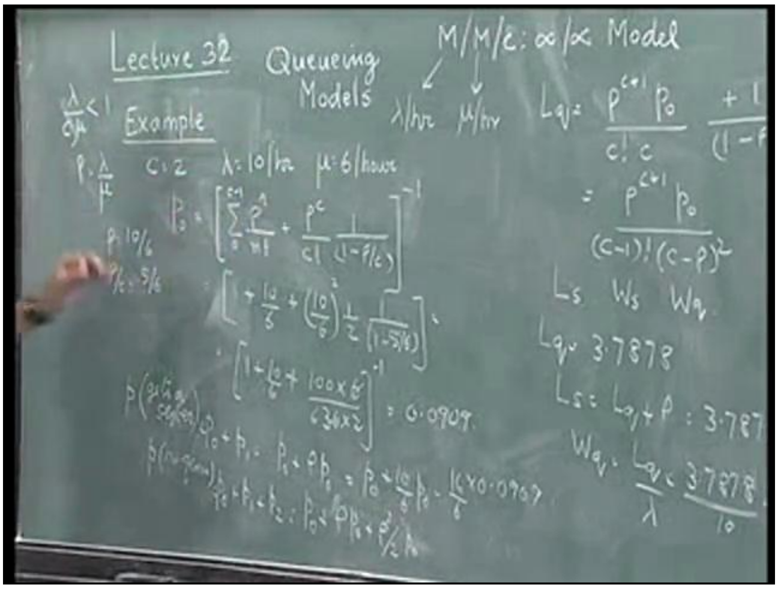
(Refer Slide Time 17:05 min)



Now what is the expected number in the system, so in order to find Ls we first need to find Lq. Lq is given by this formula rho power c plus 1 by c factorial into c 1 by 1 minus rho by c the whole square. Now we know that rho is equal to 10 by 6, c is equal to 2, so substituting here Lq will be 3.7878 and then Ls will be Lq plus rho. So this will become 3.7878 plus 1.666 which will become 5.4544.

We may also find out Wq and Ws for example Wq in this case will be Lq by lambda, which will be 3.7878 by 10, which will be .378 hour, which is roughly 22 minutes. So like this we can work out the computations to get the values of Lq, from which Wq and other expressions can be found out.

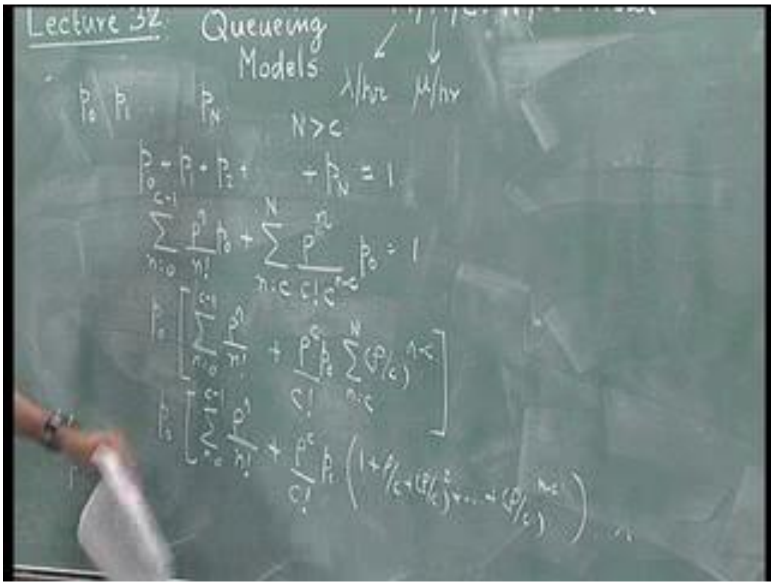
(Refer Slide Time 19:33 min)



Now the only certain difference here is that, when this is an infinite queue length model we have added one more server into this so c is equal to 2 so if we compare the old values of p_0. Now because we have more servers in the system the system will be idle much more, than it was in the earlier model we used 5 by 6 lambda was 5 mu was 6.

Now we have a system where lambda is 10 c is equal to 2 and mu equal to 6 so effectively lambda by c mu turns out to be 5 by 6. So when we have multiple servers it is only understandable that the system having more servers will have more free time available system would have more free time available at the same time it will be much more friendly towards the end user, where Ls the length of the system and Lq will reduce considerably compared to what it was in the earlier case. In the earlier case Ls was around 8 Lq was around 7 now we realize that Lq is just about half of 7 or little more than half of 7 so Ls and Lq would come down as the number of severs increase.

(Refer Slide Time 20:25 min)

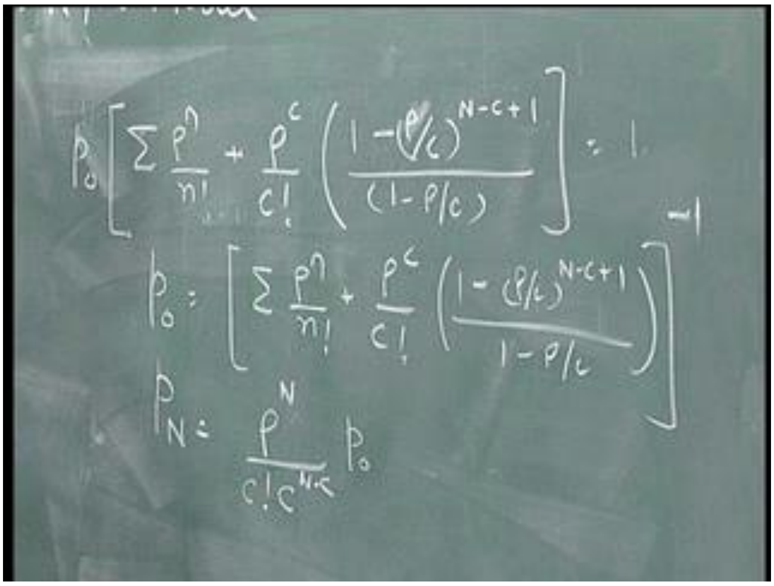


The next model that we will see, which is the fourth model is when we restrict this to N we look at the fourth model. Now the model becomes the arrivals follow a poisson distribution of lambda per hour service times are exponential with mu we have c more than one server and finite q length of not more than N. Therefore in this system we will only have 0 1 up to N people in this system it is also fair to assume that this N will be will be greater than c. So we will have a system to find out p_0 we will require p_0 plus p 1 plus p 2 plus p_N is equal to 1 and we also know that up to c the p_N expression is different and c and above the p_N expression is different so this will give sigma n equal to 0 c minus 1 rho power n by n factorial p_0 plus sigma n equal to c to N rho power. n by c factorial c to the power n minus c p_0 is equal to 1.

Now this comes, because when n is greater than or equal to c, then the effective service rate will be up to mu to mu up to c mu. So 1 into 2 into 3 up to c will give us the c factorial, after which it will be always c mu so we will get c to the power n minus c and rho to the power n into p_0. So this on summation would give us a final expression for p_0, which will be like this. Now this will give us p_0 into sigma n equal to 0 to c minus 1 rho power n by n factorial plus when we expand this we will get we can take a rho power c outside so this will become rho power c by c factorial p_0 sigma. This will become rho by c to the power n minus c see the c factorial remains as c factorial p_0 comes here. Now since we have c to the power n minus c we take a rho power c outside so that it becomes rho by c to the power n minus c this is from n equal c to N. So this would give us p_0 into sigma n equal to 0 to c minus 1 rho power n by n factorial plus rho power c by c factorial p_0 now when n equal to c we get 1 when n equal to c we get 0 when n equal to c plus 1 we get 1 and so on. So this will become, on expansion 1 plus rho by c plus rho by c the whole square plus, now, this is from c c plus 1 c plus 2 up to N.

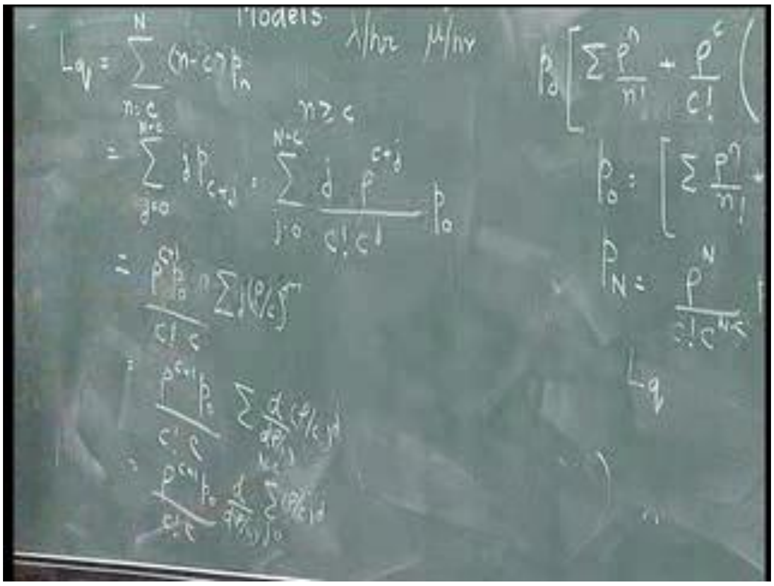
For example, if c equal to 2 and n is equal to 8 we will have 2 3 4 5 6 7 8 which are seven terms so effectively this will be rho by c to the power n minus c. First term, second term, third term (Refer Slide Time: 25:50) and for example if c equal to 2 and say N equal to 8 so when c equal to 2 this will be the term, for c equal to 3 this will be rho by c, for c equal to 4 it is rho by c the whole square so if c equal to 8 it will be rho by c to the power six which will be rho by c to the power n minus c.

(Refer Slide Time 26:26 min)



Now this is again a geometric series, but it is a finite geometric series so this on summation would give us p_0 into sigma rho power n by n factorial plus rho power c by c factorial plus rho power c by c factorial. This is a finite geometric series with n minus c plus 1 terms so this will become 1 minus r to the power n by 1 minus 1 minus r to the power n plus 1 by 1 minus r. So this would give us into 1 minus rho by c to the power N minus c plus 1 by 1 minus rho by c is equal to 1 from which, p_0 is equal to sigma rho power n by n factorial plus rho power c by c factorial 1 minus rho by c to the power N minus c plus 1 by 1 minus rho by c the whole thing inverse. So once we get an expression for p_0 we can calculate for any p_N if n is less than or equal to c, then we use this part and when n is greater than c we can use this part to get any p_N. So the only another thing that we need to do is to get Ls L_q Ws W_q. We also have the possibility of force balking, because we have a finite queue length of N. So as people get into the system if they find that already there are N people there then they leave. So we need to first find out p_N. p_N would come from here so p_N will be rho to the power N by c factorial c to the power N minus c into p_0 now p_0 is known rho is known c is also known N is known so we can calculate p_N. Now this is the probability that there are N people in the system so this is the probability that someone who comes into the system will$_0$ join and leave the system due to forced balking. Then the next thing we need to do is get an expression for Lq and then try and get expressions for Ls Wq and Ws.

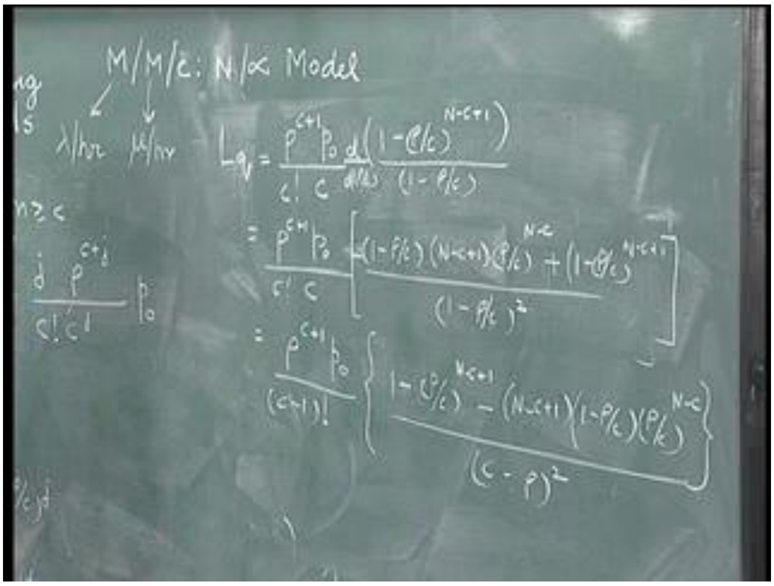
(Refer Slide Time 29:13 min)



So the expression for Lq would look like this, so we would say Lq is the expected length of the queue so Lq in this case is given by n equal to c to N n minus c p_n. So when the system has c people the queue length is 0 systems has c or less people the queue length is 0 and the queue builds, only when the number in the system is greater than or equal to c so we start from c equal to n to get this. So p_n now will always have n greater than or equal to c so we will write the second term so this will become n equal to c or j equal to 0 to n minus c $jp_{c \text{ plus } j}$. Now this will become sigma j equal to 0 to n minus c j probability that c plus j people are there in the system is rho to the power c plus j divided by c factorial c to the power n minus c, which is c to the power c plus j minus c so c to the power j into p_0.

Now again we have to come back and pull out a few terms so this will become since there is a c power j and there is a rho power c plus j we take a rho power c outside. So this will become rho power c p_0 by c factorial, then we write this in the form j rho by c to the power j minus 1 so we take another rho by c outside so that this becomes rho power c plus 1 p_0 by c factorial into c sigma j rho by c to the power j minus 1. Now this will be written as rho to the power c plus 1 p_0 by c factorial into c sigma d by d of rho by c into rho by c to the power j. Now again we switch the differentiation and the summation to get rho power c plus 1 p_0 by c factorial into c d by d of rho by c into rho by c to the power j except that we have to carefully define this summation so j equal to 0 to n minus c.

(Refer Slide Time 32:32 min)



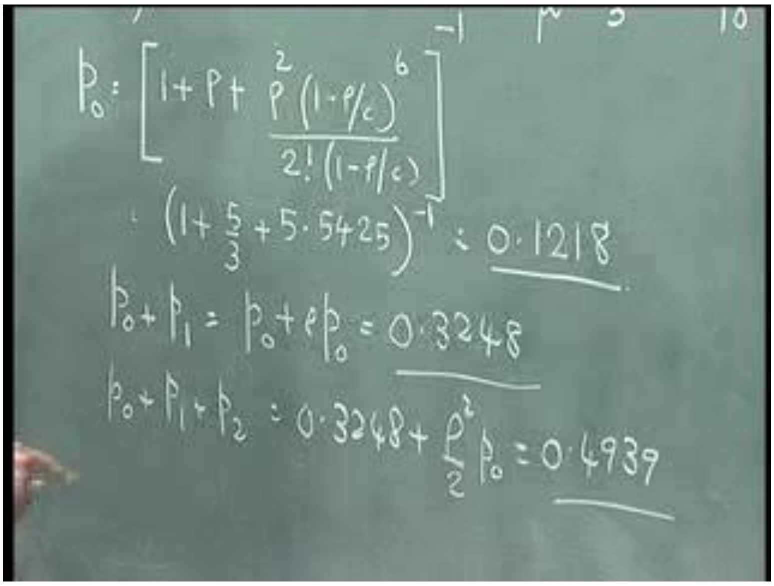
Now again continuing we would get from this L by Lq will become so Lq will become we are writing this rho power c plus 1 p_0 by c factorial into c. This is a finite geometric series when j equal to 0 this is 1 so 1 plus rho by c plus rho by c the whole square up to rho by c to the power n minus c. So we use the finite geometric series expression to get 1 minus rho by c to the power N minus c plus 1 divided by 1 minus rho by c of course we need to do d by d of rho by c. Now this would give us a reasonably complex expression, this is a u by v formula so we need to use v du minus u dv by v square. I will just show one more expansion of this and then we will write the final expression. So this will be rho to the power c plus 1 p_0 by c factorial into c this is like 1 minus rho by c so this is divided by 1 minus rho by c the whole square. Remember you are differentiating with respect to this so v du 1 minus rho by c into differentiation of this so this will go so this will become n minus c plus 1 rho by c to the power N minus c with a minus sign.

We just need to carefully put a minus sign somewhere this is a minus sign it is not minus it is a minus sign that comes in. So we will possibly do this with a minus sign minus this v du minus u dv so one minus rho by c to the power N minus c plus 1 into d of this (Refer Slide

Time: 35:02) so differentiation of this with respect to rho by c will simply become, this is like differentiating 1 minus x with respect to x so we get a minus 1 here so this becomes plus 1 into p_0.

This can be easily simplified, so this will give us the final expression which is rho power c plus 1 p_0 by c minus 1 factorial. that comes in because when we expand this we can do a c minus rho the whole c square will come to the numerator and when c square comes to the numerator 1 c will go and the c factorial will become c minus 1 factorial into, so this expression will be 1 minus rho by c to power N minus c plus 1, which comes from here minus N minus c plus 1, which comes from here into 1 minus rho by c into rho by c to the power N minus c and the whole thing divided by c minus rho the whole square, which comes out of this. So this is the final expression for Lq when we use an M M c N infinity model, so once we get this expression for Lq we can also get Ws Wq and so on.

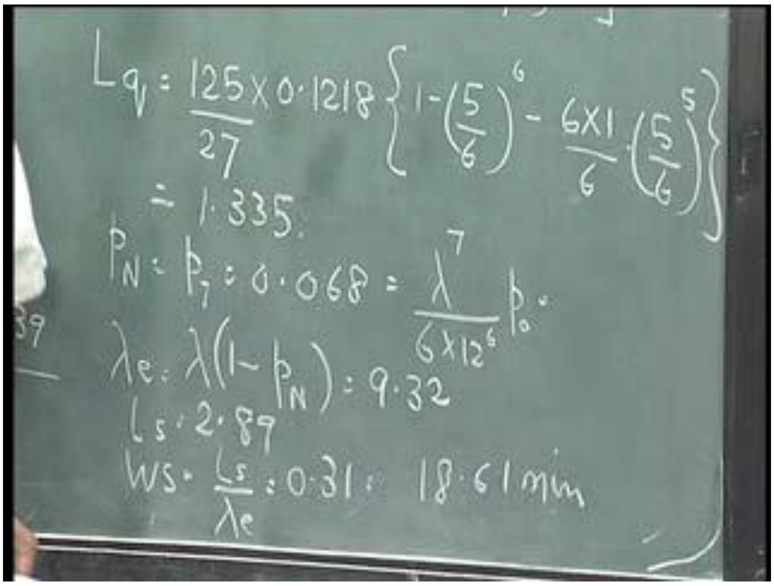
(Refer Slide Time 36:39 min)



These two are two computers and seven people in the system, which means there are five other chairs which are available. Now the probability that a person who comes in immediately takes a computer will be p_0 plus p_1 and there is at least when it is p_0 two computers are available p_1 one computer is available. So this will become p_0 plus rho p_0 and this on substitution would give us 0.3248 so there is a probability of 0.3248 that a person who comes immediately can get a system. Now probability that there is no queue on arrival is p_0 plus p_1 plus p_2, because even if there are two people in the system both these people are actually using the computer so there is no queue. So probability that there is no queue is p_0

plus p_1 plus p_2 and this will be 0.3248, which p_0 plus p_1 plus rho square by 2 p_0, which on simplification would give us 0.4939. So probability that when a person comes in there is no queue is 0.4939 probabilities that when a person come in the person can directly take a computer is 0.3248 and probability that the system is empty there is no one in the system is 0.1218.

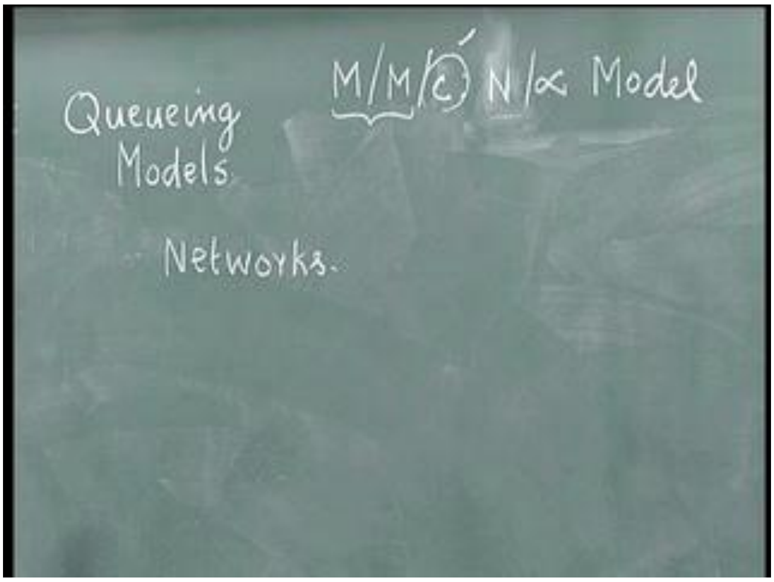
(Refer Slide Time 38:23 min)



Now the next thing that we would like to evaluate is Lq, what is the expected length of the queue at steady state for this we have this formula we know all the terms p_0 we have just computed. So this when we substitute would give us 125 by 27 that is rho power c plus 1 rho power c plus 1 is 125 by 27 c minus rho the whole square factorial into 0.1218, which comes from p_0 and then we have this term 1 minus 5 by 6 to the power six minus 6 into 1 by 6 into 5 by 6 to the power five. Now these terms come when we substitute here 1 minus rho by c to the power N minus c plus 1 n minus c plus 1 into rho by c to the power N minus c comes from this term 6 into 5 by 6 to the power five. We also have the c minus rho the whole square, which takes care of the other part here so this would give us on substitution Lq is 1.335. So, N minus c plus 1 into 1 minus rho by c into rho by c to the power N minus c.

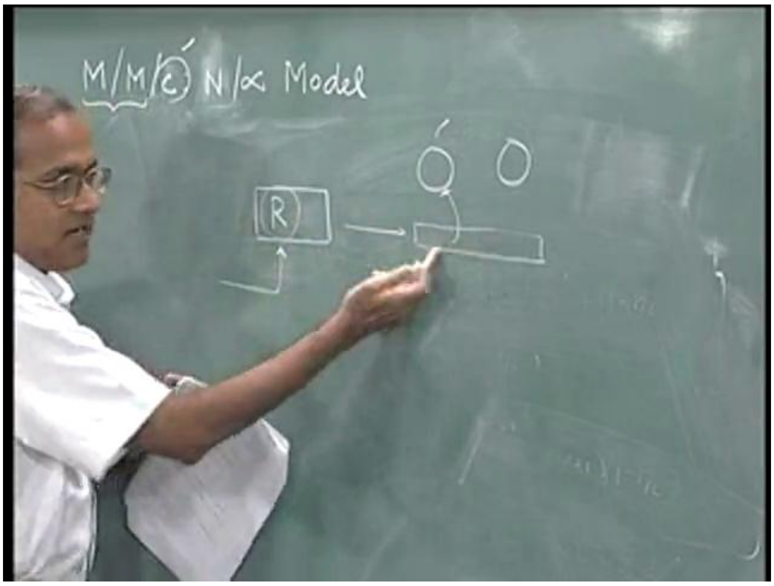
So 1 minus rho by c gives us this term of 6 so this on substitution would give us a Lq value of 1.335 this is the expected length of the queue. Then p_N is the probability that there are N people in the system, which is p_7 which on substitution would give us 0.068. In fact p_N is given by lambda power seven by 6 into 12 power six p_0 on substitution, which would give us 0.068. Now we have lambda e which is called effective arrival into the system, which is lambda into 1 minus p_N though people arrive at the rate of 10, because some people go away lambda e becomes 9.32 from which Ls will become 2.89 number of people in the system Ls will be 2.89 and then Ws, which is the expected waiting time in the system on computation is Ls by lambda e, which on substitution would give us 0.31 hours, which is 18.61 minutes. So, expected time waiting time in the system is 18.61minutes.

(Refer Slide Time 41:40 min)



Of course as I mentioned earlier there are other queueing models. The other models would mean that we can relax this Markovian assumption, we many not use the memory less property, we may have some general distributions for service and so on. So right now in this lecture series we are not looking at those extended queueing models. Now in reality there are also some queueing models which are called queueing networks. Right now we have not had a network of queue so let me first explain what a network of queue is.

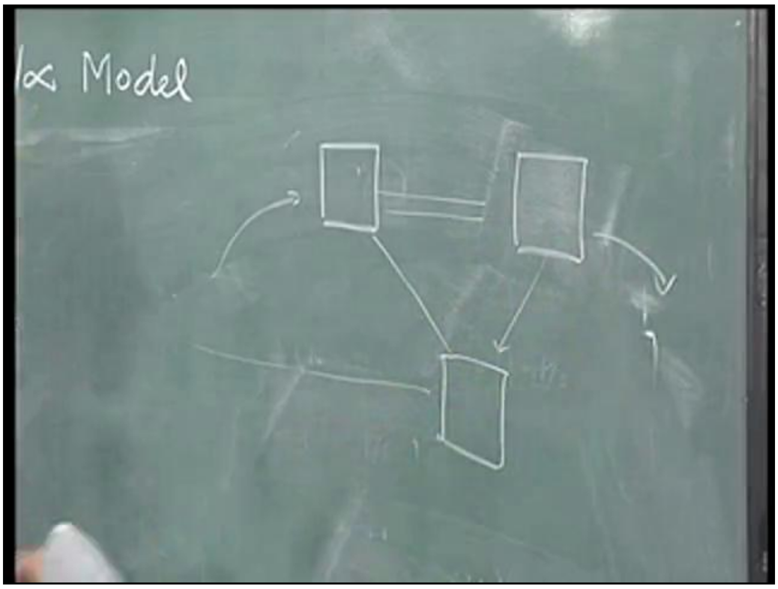
(Refer Slide Time 42:23 min)



If let us say that we go to a hospital and as soon as we enter the hospital let us say that there is a registration and then we need to go to the registration. Register our names here and then come out of this and wait in another long queue and let us say that there are doctors here and then we get served and we leave the system.

Now if you looked at a little more involved system like this there is a queue here for the registration and whatever comes out of this queue is an input. So the service of this is the arrival into this the service rate of this is the arrival rate into this and from there it expands so this is a network of queues. So when you have a network of queues the analysis are quite different we cannot assume that this is the arrival rate and then there is a service rate which comes in and so on. The equations are little more involved when we want to analyze networks of queues. But in reality most queueing systems would involve more than one service or more than one server and therefore we may have to study network queueing network models or networks of queues subsequently if we want to enrich our knowledge further in the queueing models.

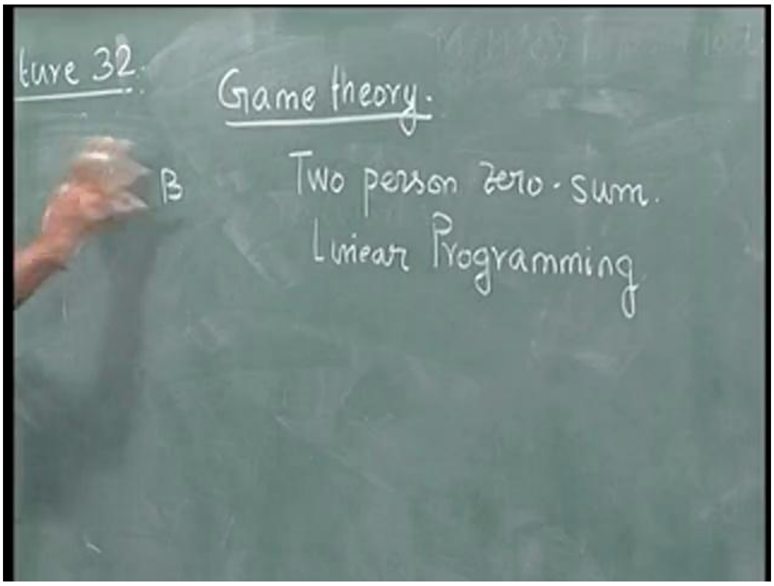
(Refer Slide Time 43:50 min)



We can also model sometimes a manufacturing system as a queueing network. Now say a manufacturing system may have certain machines and these jobs may come in here some jobs will come here and then it goes here and then they go here while some jobs will enter here it may go to another machine it may go to the third machine and go out and so on. So the manufacturing system can also be seen as a very complex network of queues. So queueing models have a lot of applicability not only in day to day services like a doctor or a dentist or a petrol station or ATM machine or a barber shop and so on but they also have applications in for example restaurants where you want to design the capacity because each seat or table in a restaurant can also be seen as a server where the time we take to eat is the service time and so on so we use these kinds of models to determine what should the number of tables in the restaurant be and so on and it also uses things like manufacturing system where the performance of the manufacturing system can be analyzed.

Of late queueing models are increasingly being used in medical systems particularly in hospitals in hospital waiting line management trying to find out the number of doctors or number of specialist that are required and so on. So there is an increasing use of these models particularly in health care related systems, which is of recent interest to researchers and practitioners. So with this is we come to the end of our discussion on queueing models.

(Refer Slide Time 45:35 min)



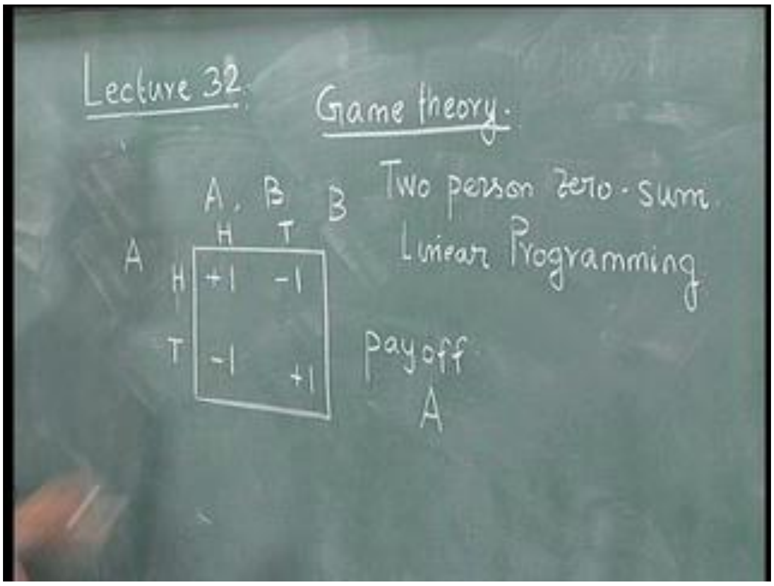
And we wish to introduce another topic here, which is called game theory. Now having completed the discussion for this lecture series on the queueing theory, we now move to the game theory and in the game theoretic models. We first describe what a game is or what a game theoretical model is and then we try and see the ways to solve the game. Now what we are going to see in this lecture series is typically what are called two persons zero sum games and we are going to see solution methodologies, which are based on linear programming. So game theory can be seen as some kind of an application area of linear programming even though if we go back into the history game theory did not or was not derived or seen as a direct application of linear programming. The theory kind of existed before the linear programming theory was established, but once the linear programming theory was established people could apply this comfortably to the game theoretic problems.

We will only see some basics of game theory define what the game theory problem is and then look at a couple of ways to solve the game theory problem. So in this advanced operations research course game theory comes in more as an application of linear programming, because we are not going to see any more theory more than what we have seen in the two lecture series of fundamentals of operation research and advanced operations research. So this would come in as some kind of an application area of linear programming. Now what is this game and what is this game theory. Now many times we are used to playing some simple games, which are like two persons sitting opposite to each other and say and are betting. Now if we take a simple situation that two people are actually playing a betting game they have a coin and let us say that they agree that they one of them tosses the coin. Assume that the coin is unbiased so when one of them tosses the coin if they agree that two people A and B are playing now any of them can toss the coin. If they agree that if they get heads then A gets one rupee or B gives A one rupee and if it gets tails the A gives B one rupee.

Now we all know that since the coin is unbiased there is an equal probability of getting a head and a tail and therefore this game will not have any profit or loss if it is played continuously for a very long period. Now if on the other hand now this game is an example of what is called a two person zero sum game because two people are playing. It is zero sum because what is gain to one person is loss to the another so the net is a zero sum so it is called a zero sum game.

Now let me expand this game a little bit and say that they play this game slightly different. Now if both of them say that each of them is having a coin and each of them does not toss the coin but simple shows either a head or a tail for example they close their hands and they can put it in such a manner that when they open it either the heads is coming or the tail is coming they do not toss the coin they are fully aware of what they are showing whether they are showing a head or a tail except that they are doing it very spontaneously. Now if they say if they agree between themselves that if both of them show heads or both of them show tails A gets one rupee, whereas if the pair is a head tail or a tail head then B gets one rupee. Now let us try and match this game.

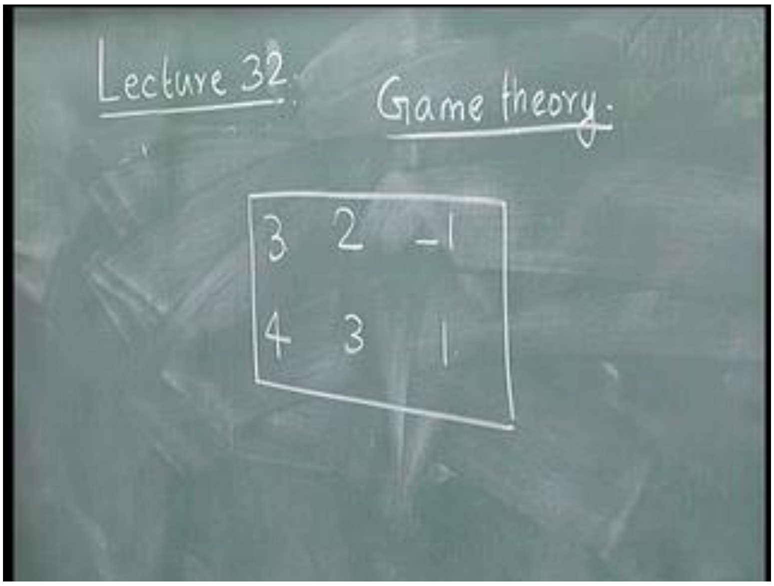
(Refer Slide Time 49:40 min)



Now, to express this game in the form of a small table I am going to assume that this is person A and this is person B. Now when this game is being played, now A whenever he has a trial where he shows a coin A can show either a head or a tail .So A has two ways of doing it, so A will have say I would say one strategy for A is to show heads the other strategy for A is to show tails. Now B also has similarly two strategies so B can show heads and B can also show tails. Now there is a payoff, as we said in the terms and conditions of the game that if both happens to be heads A gets one rupee so let me put a plus 1 for A here. We also said that if both are tails then A gets one rupee so let us put a plus 1 here and we also said if it is a head and a tail or tail and a head now B gets one rupee which means A loses one rupee so I put a minus 1 here and a minus 1 here. Now this is called the payoff matrix for A because this is not the payoff matrix for B this is the payoff matrix for A because A knows that if I play heads he is he plays heads I get one rupee if I play heads he plays tails I lose one rupee so this minus one comes here.

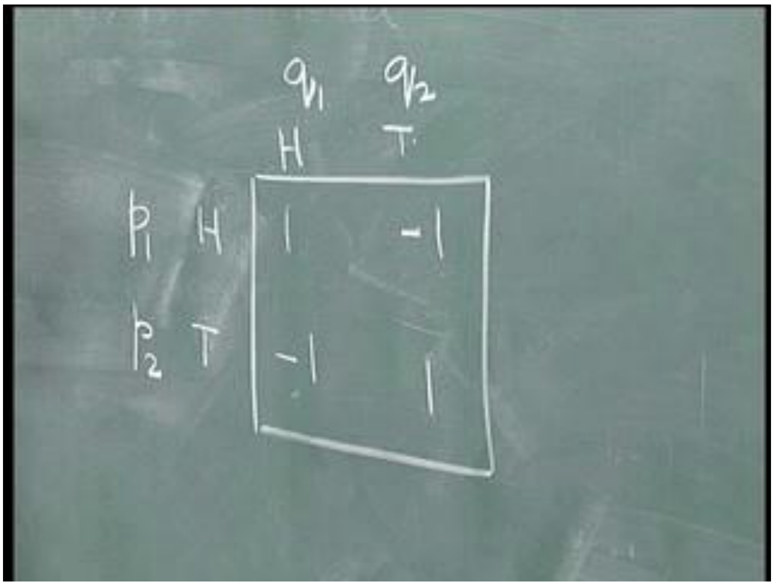
Now this is an example of a payoff matrix, but payoff matrix for A, because it is a 0 sum game the payoff matrix for B is automatically defined it will be minus 1 1 1 minus 1, because it is a 0 sum game. So whatever is a gain to A is a loss to B. So we need not define a payoff matrix for B by defining the payoff matrix for A we are automatically defining the payoff matrix for B. It is not absolutely necessary, that this has to be only a 2 by 2 matrix this can be even a general n by m matrix where A may have more than one strategy B also may have more than one strategy and then we have a payoff matrix. So let me show a payoff matrix like this so let us consider another payoff matrix.

(Refer Slide Time 52:23 min)



Now let us say A has two strategies now which you may call 1 and 2 B has three strategies 1 2 and 3. Let us say this payoff matrix says that if A plays this strategy B plays this strategy A wins three rupees. If A plays this strategy B plays this strategy A wins one rupee. If A plays this strategy B plays this strategy A will lose one rupee and B will gain one rupee. Now we are also going to assume like in the coin example where we had this coin example.

(Refer Slide Time 53:15 min)



We said 1 minus 1 minus 1 1, now A has two strategies in this to play either a head or a tail please remember that A can deliberately play a head show a head or show a tail. Now how do you expect A and B to play this let us assume that they are playing this for about hundred times or two hundred times or a large number of times. Now what will happen is A might start by saying let me play heads, because I am getting one here. So A will start playing heads and after a while B will choose between heads and tails and after a while B will realize that if A is consistently playing heads it is unwise for B to play tail to play heads so B will start playing tail so that B wins and the moment B starts playing tail consistently A will understand that A should not play heads then A will switch to tail so that A wins. The moment A switches to tails B will now start switching to head so that B wins and so on. So as the game progresses they play these strategies so many times and such that there is some steady state gain or loss at the end of the system

So each of them would actually like to find out in some sense in what proportion or in what probability they play these strategies. So now A would be interested in saying if p_1 and p_2 are the proportions or probabilities with which I can play these strategies what should be the values of p_1 and p_2 against some objective function. Similarly if B is going to play this with probabilities q_1 and q_2 or with proportions q_1 and q_2 what should be these proportions such that a certain objective is achieved? So in some sense we have to find out the steady state probabilities with which A will play these two these two strategies as well as B will play these two strategies. Now both of them are looking at playing these at certain proportions, so that they try to optimize a certain objective function. Now obviously A will try to maximize his game, because that is what A would like to do and B would like to minimize his loss, because that is what B would like to do. There is a slight modification to these objective functions and this modification and some more aspects we will look at it in the next lecture.