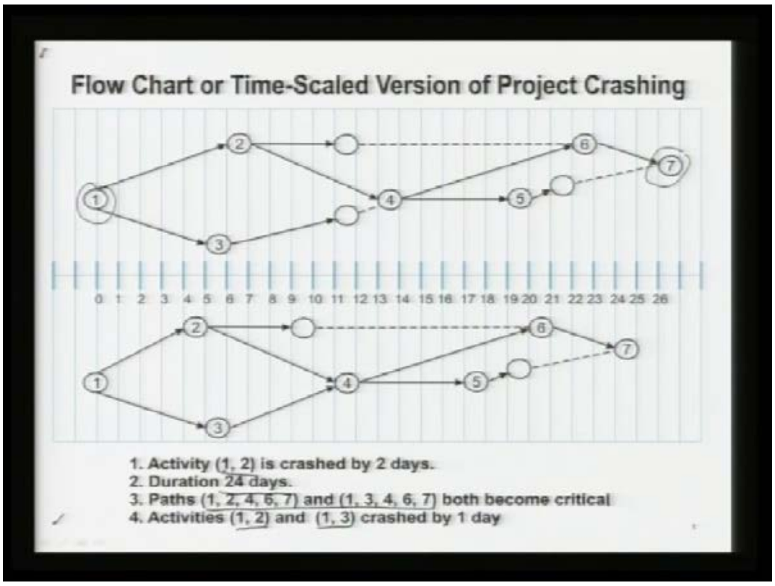
Linear Programming and its Extensions
Prof. Prabha Sharma
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture No. # 38
Programme Evaluation and Review Technique (PERT)

The example of project crashing, that I had discussed in the last lecture, I had shown you the calculations from day to day and the computed the cost and found out the optimal duration of the project for you. Now, I thought, I will also show you the flow chart version because most of the time, the management likes to look at the whole thing, you, as a one, you know, full view of the project and then monitor it from day to day.

(Refer Slide Time: 00:37)



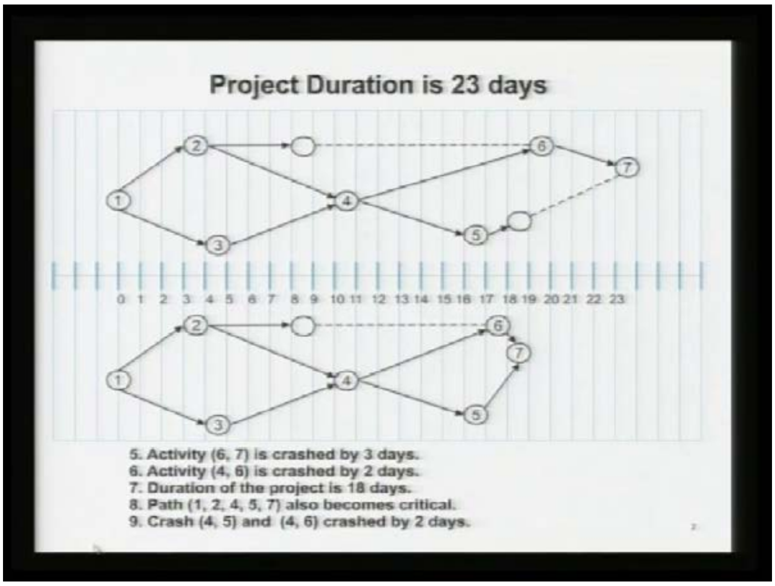
So, the, they have the whole flow chart on the one wall, covering one full wall and then, they try to keep on updating the progress of the project and so on and so I thought, you should also be aware of this. So, now, that given project, you see, that here you can have a full view of the project.

From here, as I am showing you is start. Starting of the project is indicated by node 1 and ending of the project is indicated by 7, 0 is the starting days. So, now, you can also see from here, what are the activities, which have slack time. For example, activity 2, 6 has a slack time; activity 3, 4 has a slack time and activity 5, 7 has a slack time. And you can also notice that the critical path is 1, 2, 4, 6, 7; this is the only one, which in, on which nubmer activity has a slack time.

So, activity 1, 2 is crashed by 2 days, you can see that and then, the project duration becomes 24 days, you can see from here also. And then, once that happens, paths 1, 2, 4, 6, 7 and 1, 3 both these paths become critical. So, therefore, now we need to crash activities on both the paths and it turns out that activity 1, 2 and 1, 3, I have shown you the calculations, these have the minimum rate of crashing. And I can crash them by 1 day because after this, you see, 1, 2 has a slack time of 3 days. So, I have already crashed it by 3 days, therefore I cannot crash this combination by more than 1 day, but I do, I use, I choose this combination because this has the, a smallest rate of crashing.

So, once I do this by one day then, the project duration will become 23 days, which I will show you in the next slide.

(Refer Slide Time: 02:17)



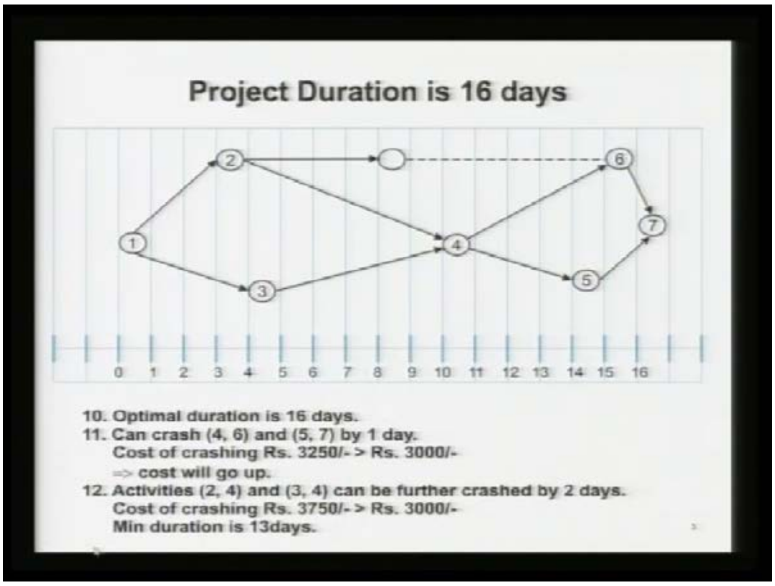
So, project duration is 23 days, which you can see from here and you still have activities 2, 6 and 5, 7 having slack, but 4, 5 is no longer having a slack and therefore, sorry, it was 3, 4 I think, so anyway.

So, therefore, you had 2 paths become critical 1, 2, 4, 6, 7 and 1, 3, 4, 6, 7. Both these paths became critical, as I pointed out in the earlier slide. And so now, I need to crash activities, which are common to both the paths or which will reduce the path length of both the critical paths, then only the project duration will get reduced. So, you see, that activity 6, 7 and because this has a smallest rate of crashing, is crashed by 3 days and activity 4, 6 is crashed by 2 days. And so the total duration, therefore, gets deal reduced by 5 days and so the project duration is now 18 days, you can see this here.

And now, you see, that the activity, the, yeah, so the slack for activity 5, 7 is also gone, earlier it had a slack of few days. That means, it had actually slack of 1, 2, 3, 4, 5 days. So, now, that is also gone because I reduced the project duration by 3 plus 2, 5 days. So, now, you have 3 paths. So, 1, 2, 4, 5, 7 also becomes critical. So, you have 3 paths, which have become critical and again.

I need to crash combination of activities, which will reduce the duration of all the 3 critical paths to help reduce the project duration. And so, you see, if I crash 4, 5 and 4, 6, 7 by 2 days, so 4, 5 and 4, 6 by 2 days, 4, 6 had (( )) slack, so then, project length will come down to 16.

(Refer Slide Time: 04:06)



So, it comes down to 16, you can see that here. And then, what happens is that after this, if I want to crash the project duration further, when I feel, that I can choose this 4, 6 and

5, 7 because 4, 6 has only 1 day slack left. So, therefore, this is the pair and because this has minimum cost among all the pairs, that you can reduce.

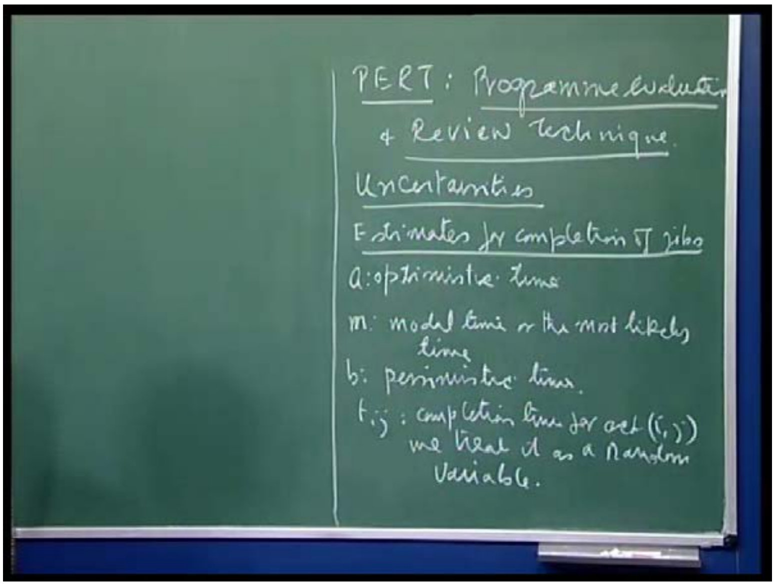
So, but the cost of crashing now is rupees 3250, which is more than the overhead cost. That means, your reduction will be 3000, but your cost of crashing will go up, the direct cost will go by 250 more. So, therefore, we know that 16 days is the optimum duration of the project because now any set of activities, that you call crash, the rate of crashing is more than the overhead cost. So, therefore, the cost will go up.

Similarly, you can also crash activities 2, 4 and 3, 4 further; 2, 4 and 3, 4. But here again, the cost of crashing is more than the overhead cost. The cost of crashing is 3750, which is more than your overhead cost of 3000 rupees per day, but if you want to know the minimum duration, because sometimes you want to know how far can you go, how much can you reduce the project duration, so you crash it by 2 days, and that will make the project duration is, 15, 13 days. So, I am calling it minimum duration because now you see, only activity 2, 6 has a slack, no other activity has.

And you know, reducing this will not help me reduce the project duration because the critical paths are all here, this is not part of the critical path. So, therefore, I will call this duration of 13 days as the minimum duration and that is it.

So, you now have a, you know, a good view and you can, you know, for by looking at the progress of the chart, you can you know monitor the project very well, this is the idea behind.

(Refer Slide Time: 06:07)



So, the next topic in the same line, I want to talk about is Pert, this is programmme evaluation, evaluation, and review technique.

So, the idea here is and again we can say that the CPM and Pert are tools for handling large projects. For the, c the, CPM, situation was when the, when you are, sort of very, almost confident about the resources, the duration and so on, of the activities and so you have a fair idea to how long the whole project will take and how long each activity will take.

But Pert handles the situation, or it comes in handy when it is a new project, may be, say for example, our space center, they have been building these Chandrayaans and so on, so this is the completely new project. When they sent Chandrayaan-1, it was a new project and so there were lot of new things, which they had not done before and they could only go by estimates. And of course, even a single estimate may not be enough. So, this is what. So, therefore, when uncertainties are there, so this comes in when uncertain of, this handles certainties.

So, and when situations are uncertain, you are not confident or you have not enough knowledge about the projects, about the jobs involved in this, in the, in the project and the project is huge, then Pert comes in and this is a very useful technique and we try to look at it. So, here, the idea here is that you may, you may ask for, so estimates, you may ask for estimates for completion of, of jobs, estimates for time, if the estimates for jobs, job completion time.

So, (( )), is defined here, a is, a is optimistic because a is the shortest duration. So, you say, that if the person is very optimistic, that everything will go as planned and so on. So, then, a, this denotes the optimist, optimistic time. Then, m would denote the model time or the most likely time, most likely time, which, that means, you take care of some failures and something is going ok and so on. A mixture of, so this, you, the, the person concerned with the particular activity or job gives you the, the estimate, that ok, this I think would be the most likely time. And then, b is the pessimistic time that means, when everything can go probably; so, something like that.

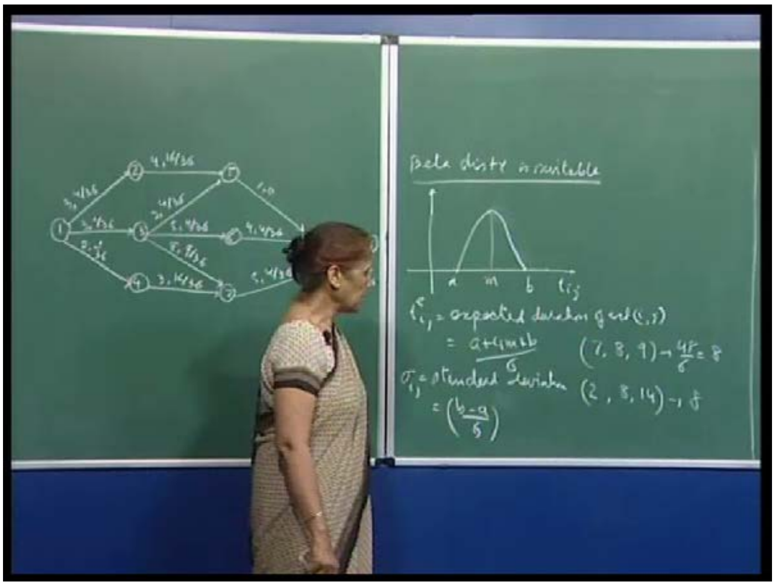
So, these are the 3 estimates given and since there are uncertainties, so what we would say is that the completion time for each activity is the, is the random (( )); so, therefore, random variable. So, we will say, that your t ij, this is completion time for activity ij. So, accordingly, these 3 numbers will be a ij, m ij, b ij for activity ij. So, if t ij denotes the completion time for at this, then we treat it as a random variable; we treat this as a random variable.

And obviously, the burning question is that what a, what is, what are the probability of whether project being completed by a certain date and so on? So, you want to compute that. So, that means, since we are treating t ij as a random variable, you need a probability density function and you see the characteristics of the probability density function, which describes the random variable t ij is what, that it has a finite range because it should be, see you do not expect the job to be completed before the time a, because it is the optimistic time, and you do not expect it to go beyond b. So, that means, it is a.

So, you are looking for a probably density function, which has a finite range, which is unimodel; unimodel means, that it does not have more than 1 hump and then, it is not symmetric, it is continuous, it is not symmetric because the activity can be, if you are not saying, that it has to be integer values and so on. So, therefore, you are looking for density function, which is, which has a finite range, unimodel, continuous and not symmetric.

So, let us continue with the problem. So, here I was describing, that we need probability density function to describe the random variable t ij, which is finite in range and which is unimodel and which is continuous and asymmetric.

(Refer Slide Time: 11:54)



Beta distribution is taken as the, so beta distribution describes, distribution is suitable, is suitable for the situation and as you can see, that an idea here is, if this is your a and this is your b, then something, it should not be, it should not look so symmetric, I should try to make it a little asymmetric, something like this. And so, this will describe your m, this will be your m; so, this is your t ij, and this is the, well, this is the time and this describes the, is a probability of t ij being, were taking a certain value in this interval. So, this describes the curve.

So, this is your beta distribution and lot of work has been done, so we really do not have to worry, why? This is suitable, as I told you, the main points are, that this is, this is a finite range because most of the other probability density functions, that you know are, either I was, say for example, a normal distribution has range from minus infinity to infinity, the gamma distribution has a range from 0 to infinity and so on. So, this is o.k.

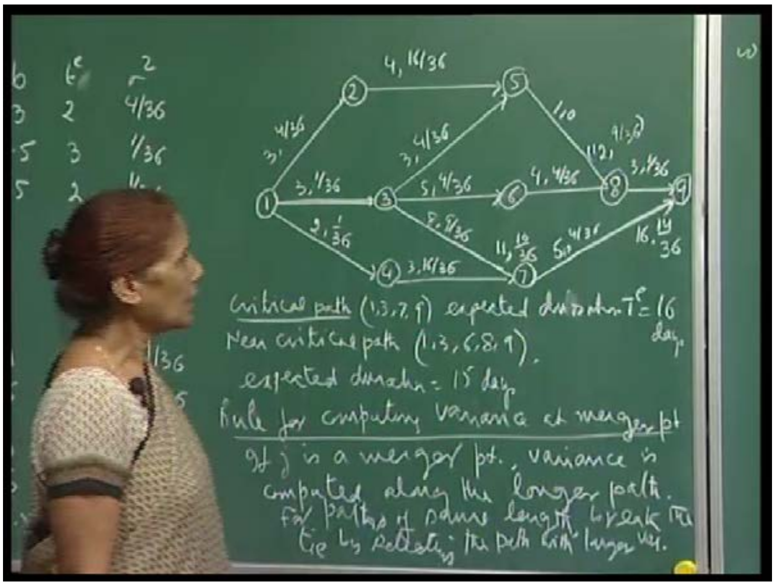
And now we make an estimate. So, what we say is that for example, expected, expected duration of activity ij, the formula is a plus 4m plus b upon 6. These are a good fit, is supposed to be (( )).

So, expected duration of activity ij is, computed approximately by this formula. And the, and now see, what happens is that, this is not enough because if you have simply just the expected duration of an activity, let me give you numbers, say for example, if you look at the number 7, 8 and 9, that means, a is 7, m is 8 and this is 9, then the expected duration would be what? According to this formula would be 7 plus 9, 16; 16 plus 32, 48; so, 48 divided by 6 will be 8. But if you take the numbers, let us say 2, 8 and 14 and I can, I can make the spread even more bigger, I have just chosen this. So, this also has the expected duration as 8.

But certainly, you can say that here, you see the time estimates for the activities are very close, 7, 8 and 9 differ by one day only. Whereas, here, the difference, the span is the, or the range is from 2 to 14; that means, 12 days span. And so, the expected duration of an activity is not enough because if you want to derive anything from the data, then we also need the variants. So, the formula for sigma, that is, so ij standard, or I should have written here a ij and so on, but standard deviation, so the standard deviation is b minus a by 6 whole square.

So, these are the 2 formulas and we will see, that why we need. So, no square if I am not writing this square here, this is the standard deviation and if you square the number, it becomes the variance. So, we would need both, the expected duration of the activity and the variance of the activity. So, once you have computed the expected iteration, then we will treat in some sense, we will treat, that the, the computation there, that the completion times are deterministic in the sense, that I have this expected duration of the activity.

(Refer Slide Time: 16:08)



So, yes, I just want to discuss this example with you for, for Pert and what we are saying here is that yeah, the activities are the a, m, b are given, then we have computed the expected duration of each activity and the variance for each activity, which, which are shown along the arcs here. And then, we need to, so, and then of course, we will then take the expected duration of each activity as the given time for completing the activity. So, then, I can use the critical path method and find out the longest path here or the critical path.

So, it turns out to be, yeah, the critical path comes out to be 2, 3, 5, 5 and 5. What is my critical path? 1, 3, 7, 9; so, this is the critical path, yeah, you will use exactly that method that we discussed for. So, this is your critical path, 1, 3, 7, 9.

The expected duration of the project is 16 days and the variance is 14 by 36, you add up the variances on the each…, See, the thing is, that I, in the beginning I told you, that we, we breakup the project into activities, which we think are in some sense independent of each other; that means, the resource requirement and so on and the time taken to complete the…

So, you can break up the project into small activities and we assume independence. If you assume independence, then you can remember, that you can add up the, if you have to sum 2 variables, x plus y, 2 random variables, then the variance will be variance of x plus variance of y under independence.

So, this is what I am doing and see this would be more realistic when you have a very large project. So, here, just to show you the methods, I am taking a small example and but normally, you will have a few hundred jobs and so the independence would be more realistic in that situation. So, anyway. So, then, I will just add up the variances of the activities, which make up the critical path. So, that gives me, 3, 1 plus 9, 10; 10 plus 4, 14 upon 36.

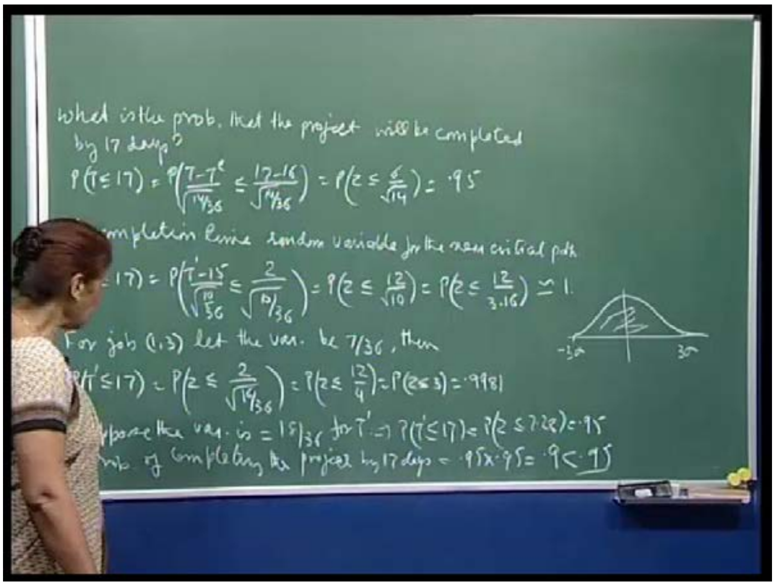
The thing is that at merger points what happens, see for example, if you look at this point here, 7, 2 paths are merging and if the lengths of the 2 paths are the same, then you will break the tie by taking the path, which has a larger variance because getting the activities completed on a path, which has larger variance, the uncertainties higher than for a smaller variance.

So, if there is a tie for the 2 lengths of the all the paths, which are merging at a, merging, merger point, then you will choose the path through the larger variance, but if, but otherwise, you will always compute the variance for the, along the longer path. So, for example, at 7 this path length is 11 and this is only 5. So, even though the variance here is 17 by 36, I will go by this variance 9 because the difference between the 2 path lengths is, is, is significant, it is only 5 days, whereas here it is 8 day, 11 days. So, that is the rule.

So, we are saying, that if j is a merger point variance, is computed along the longer path and of course, for, for paths of same length, same length, break the tie, you, just, I am speaking out break the tie by selecting the path with larger variance, larger variance. So, you are not selecting the path actually, you are selecting the variance of the merger point; so, that would be along the longer paths. In case, there is, in case there is a tie, it will be the variance for the merger point, will be the larger variance.

So, that, that of course, is not happen here. So, by this rule you will compute, you start computing the variances of, for these nodes, for these nodes and so on. So, you reach up to node 9, which is 14 by 36.

(Refer Slide Time: 20:45)



So, now if you want to, so the kind of questions, that you normally, the management has to answer, so one of the question would be, what is the probability that the project you will be computed by 17 days.

Suppose, whoever wants to get the project completed has given you the deadline, so then, you have to compute this probability. And see, here again we will, because we said, that each activity of the project has a beta distribution, so then, and they are independent. So, the central limit theorem says that if you have a large number of independent random variables having the same kind of distributions, then they are some, will behave like, in approximately will behave like a normal distribution. That means, some of the random variables will, the approximate distribution will probably, distribution would be a normal distribution. So, therefore, here, now when you say t is the completion, where t is the random variable, which denotes the completion of the project and since we are computing this along the critical path, so then we are saying, that these, some of these random variables, which denote the completion of separate activities, that add up to your t random variable and this is like a, this has a normal distribution, it has a normal distribution with the expected duration of the project as the mean and the variance has the sum of the variances.

So, this is the idea. So, we are using the concept of a central limit theorem and then, I can standardize this variant. So, the t minus t e upon the standard deviation this gives becomes a normal variant with mean 0 and variance 1. So, then, and standard deviation 1, so this is the number, so actually, you are computing this and this you can, you know, with their calculator find out this number and then looking at up the table, this comes out to be 0.95. That means, what you are saying is that in your this thing, standard this thing, this number maybe somewhere here. So, this is the area, which is equal to 0.95.

So, if this number is 6 upon root 14, so the tables will give you the area under the curve up to a certain point, so that is 0.95. So, let it be just a good probability and therefore, the management with confidence can say, yes there is a probability of 0.95, that the project will get completed by 17 days, but then, see what happens is if you look at the near critical path, it has only expected duration of 15 days and this is 1, 3, 6, 8, 9. So, what is the variance here? 5 plus 4, 9; 9 plus 1, 10; so, the near critical path, the variance is 8 and 3, 11; and what is happening, 4, 4, 8; 8 and 1, 9; 9 and 1, 10. So, this is 10 upon 36.

So, I am trying to show you, that once we have completed the, we have computed the probability of completing the project by 17 days. So, that came out to be 0.95.

Now, if I want to look at the effect of the near critical path. So, let us see. So, t prime is the completion time random variable for the near critical path. So, then, we want to compute the probability, that t prime is less than or equal to 17. So, this again, by the same calculation will come out to be this because the variance for the near critical path is 10 upon 36. So, therefore, this comes out to be this, which is this. Now, 12 upon 3.16 this probability is close to 1 because you know, that for the normal curve the, to almost 0.99%, 0.99 of the (( )) lies between minus 3 sigma and 3 sigma. So, if this number is more than 3, then see this probability will be equal to, almost equal to 1, so 0.998, something like this.

So, therefore, right now, because the variance of this near critical path is small, so this will not have any bearing on the probability of completing the project, but then I was trying to show you, that if you suppose increase the variance of job 1, 3 by, you know, make it 7 by 36, then, then still this, this will become 16 by 36, the variance of the whole near critical path and therefore, this probability will again be 3, so which is 0.9981. So, therefore, again, in fact, this number will be much more than 3. So, anyway, this probability is equal to 1, but here this one is also 0.9981.

So, therefore, again, because the, the, so essentially, I am trying to say is that the near critical path is close to the critical path and the variance is high, then it happens, that the probability of completing the project will not be just dependent on the completion, completing, completing the activities on the critical path.

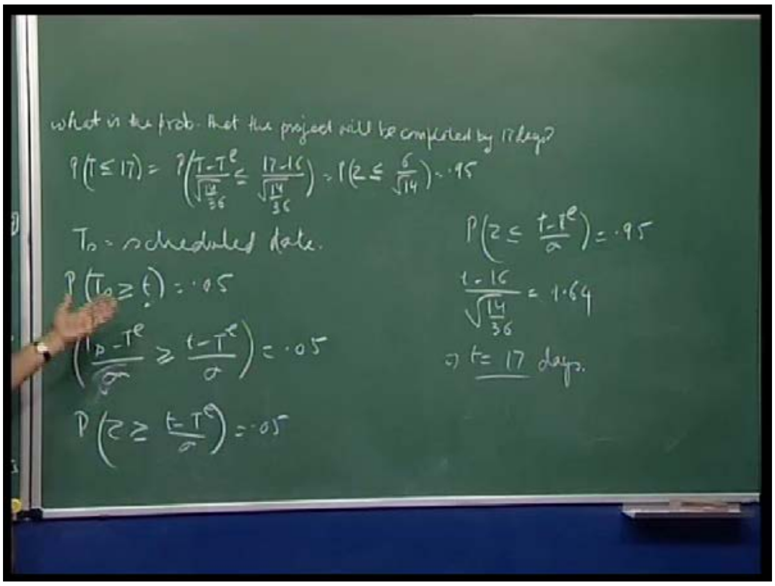
So, I am just again hypothizing and saying, suppose the variance is actually 18 by 36 for the near critical path. So, in that case, this probability will come down to 0.95 of completing the activities on the near critical path by 17 days, would be 0.95 and in that case, you will now say, that you to give you a good estimate for completing the project by 17 days, you will multiply the 2 probabilities because all the activities on the critical path and near critical path should be kept, should get completed by 17 days. So, when you multiply 0.95 with 0.95, then it will be 0.9, which is less than 0.95.

So, essentially, it is all a question of the manager having a good confidence about completing the project by a certain date because there is so much uncertainty in the project. So, this is what, what I am trying to say here is that you would have to sometimes take into account the near critical paths also in trying to find out the probability of completing the activities on the near critical path by a certain date.

Now, another kind of question, that you may be required to answer is, you know, like this is a very good tool for, for a manager to…, For example, if you are bidding for contracts and since the situations are always very competitive, you want to make a very realistic bidding because the penalties for not meeting the deadlines are often very high because these projects are very, very, you know, there are, they are big projects and lot is at stake. So, you have to be very sure of it, with lot of confidence, only you should be able to bid the project, and therefore, you would want to tell a probable date by which you think. So, for example, if you want to answer the question, yeah, let me just write it. That means, your, you are saying, that what is the probability that your project will get completed by a certain date, which is at, say for example, here again I can say, that 17 days.

So, if you want to know the probability. So, yeah, I can, I can, the question asked is what is the value of T s.

(Refer Slide Time: 28:01)



So, your T s is the scheduled date, scheduled date. So, the question asked is, what is the value of T s, so that there is only 5% chance of exceeding the contracted date? So, therefore, then you will ask a question, probability T s greater than or equal to T?

So, you want to know, this is 0.05, you want to know, that by a certain contracted date your scheduled date should be the probability, that this will exceed the contracted date should not be more than 0.05. That means, the probability of completing the project will be, then the probability would be 0.95, so therefore, this again, you will say, probability T s minus, yeah, so the same thing, T e upon under root of, we are writing sigma here for the, which is greater than or equal to T minus T e upon sigma, this should be 0.05. And this imply and...

So, this is the standard variant, so you have, saying probability z less than or equal to T minus T e upon sigma, this should equal to 0.05. So, from the normal tables, essentially, z less than or equal to this number is 0.95, you can, whichever way you want to read this thing, which is, so here in, in our this thing, we are saying, that this probability z less than or because the complement of this is less than or equal to t, small t, minus T e upon sigma, this is 0.95 and you know, that for your T e was 16, your sigma was root 14.

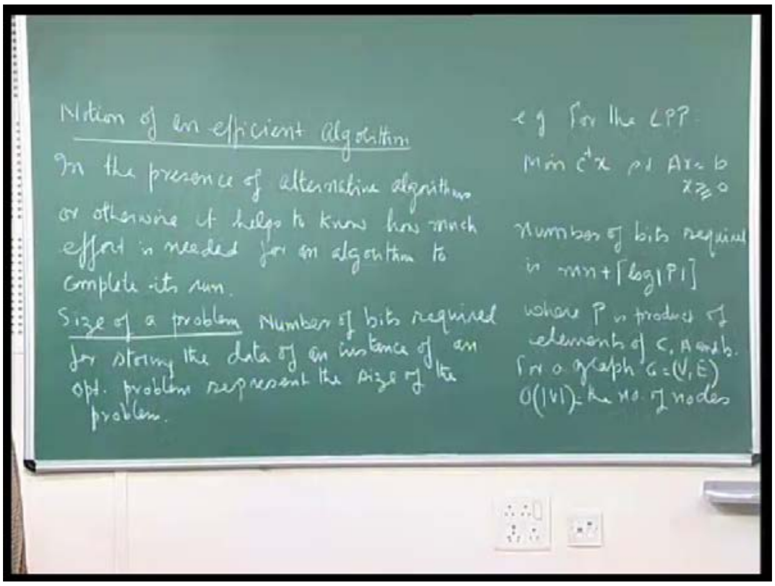
So, therefore, you are saying, that t minus 16 upon root 14 by 36 is 1.64, comes out to be from the normal tables, this is 1.64. So, this implies, that your t is equal to 17, just make the computation. That means, you will multiply by this plus 16 and this is 17, so 17 days. Of course, I took the same numbers, but you could have changed this thing, for example, you could have made it as, 0.3, 0.03, in that case, this number will come out to be higher. Because if you want the chance of failure to be much smaller, probability of failure to be less than around .03, then this will come out to be more than 17 days, but…

So, you see, again, as I have been saying right from the beginning, that everything is a sort of approximate, nothing is certain because you are, anyway, dealing with the activities, which have lot of uncertainty built into them. So, but this definitely gives sort of a guideline to the management or to the people who want to bid for contracts to get projects completed, as to how, you know, things will work out and with what probability, which sort of gives you small, some confidence as to how to handle this thing.

So, I think this should give you a feeling as to how to go about calculating the expected duration, the variance and then, what we mean by the expected duration of the project, which means the expected completion of the activities on the critical path and then, we want to answer these questions as to, what are the probabilities by which you can, certain date the project will get completed and so on.

So, with the limited time, I suppose, they should give you an idea and then, of course, you can read up wherever you need any more information.

(Refer Slide Time: 32:12)



So, before I talk, start talking about interior point methods and you know, about, I have been, in the, in the, in the previous lectures I have been sometimes referring to the amount of time, that a particular algorithm will take and the, this was, you know, in an informal way, I was trying to give you an idea as to the kind of effort, that will be required for, we know, for an algorithm to complete its run. So, I will, in this lecture, now give you some formal definitions and make the, make the concept of the complexity of an algorithm formal thing.

So, now the, first we need to have this concept of, there is a notion of an efficient algorithm. So, in order to define what we mean by an efficient algorithm, I will have to make a few definitions and the whole idea is that you know, why we need this notion, they were many reasons of course.

If in the presence of more than 1 algorithm for a particular problem, which can happen, then you may like to be able to compare the algorithms and, and then, you need to know, how much effort is required for each of the algorithms? Before, you, so that can be one yard stick for comparing the performance of 2 algorithms. Also, otherwise it helps to know, whether what kind of an effort will be required by an algorithm and sometimes you can classify, that the algorithm is such, that the amount of time required is, is not, is not a reasonable amount of time, like if one has to spend life time trying to get to an optimal solution, then of course, it is not worthwhile. So, they are lot of considerations when you want to look at the performance of an algorithm and you want to analyze it.

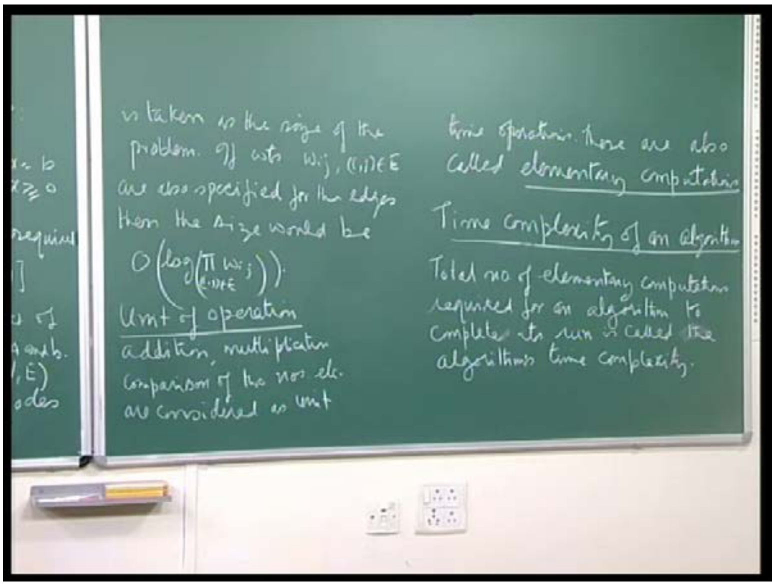
So, to do that, we first say, that the 1st thing I need is, what is the size of a problem? So, the size of a problem is number of bits required for storing the data of an instance of an optimization problem, represent, so that represents, so the number of bits, that are required for storing the data represent the size of the problem.

So, the, the idea here is that in a computer and of course, this is all reference, with reference to the performance of an algorithm when programmed to run on a computer, so they, for example, for the linear programming problem, minimize c transpose x subject to a x equal to b x non-negative. Then, you need to store the parameter c, a and b, the m by n matrix. So, then, what you do is, here, so this m, n means, that you would need that many separations. In fact, to be more exact, one can make it m plus 1 because you will have the m rows and then you will have 1 for c transpose and then, so and you could make it n plus 1 because n columns and then 1 column for b. So, this could as well, I have been m plus 1 into a, but it does not matter, it will only be a matter of m plus n numbers.

Then, you say, that because your, remember for the computer the arithmetic is discrete, so the log to the base 8 is one of the ways of recording the data. So, then and p is the product of the elements of c, a and b, so all 2. So, when you put log, so that means, all these numbers we have to record and so that will be the size of the, of a linear programming problem, of an instance of linear programming problem this would be the size.

Now, for, for problems, which are defined on graphs, if you have g equal to v e, where v is the node set, then normally we say, that for, for storing graph, that means, the nodes and the adjacent.

(Refer Slide Time: 36:14)



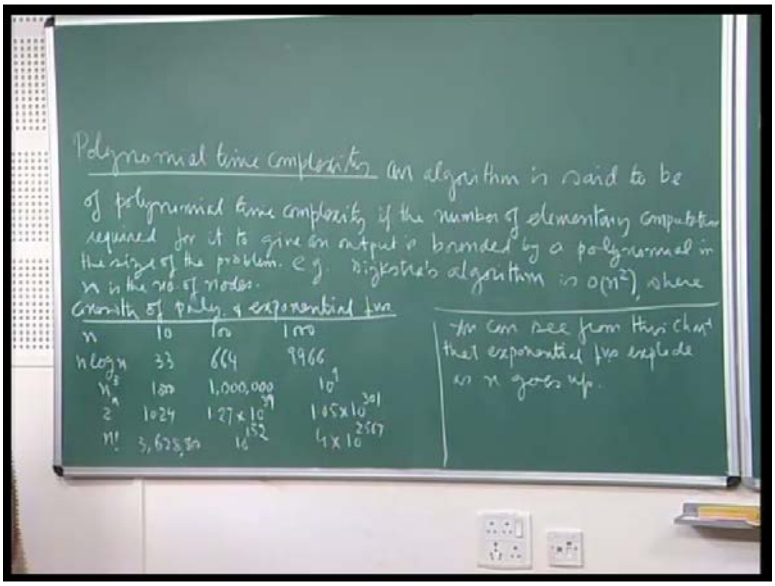
We think, then you need order v, the number of nodes, which does this, what we say is the size of a graph and if you have weights attached to the edges of the graph, then you would have to, the size would be taken as order log, product of all the edges, all the edges weight, edges w ij. So, pi w ij, ij belonging to e log of that.

So, the size of the problem, that means, for example, you are looking for shortest path problem or max-flow and so on. Then, you would have weights attached to the edges and then, you want to, the size of the problem would be something like this. So, and as, while I was discussing some algorithms, I did give you an idea as to what we mean by the size of the problem.

The 2nd thing, that we need is, what is the unit of operation? And this is like, addition multiplication and of course, addition, then subtraction also becomes part of it; multiplication implies division also and the comparison of 2 numbers etcetera. So, the, this list is not exhaustive, there will be many more. So, we call these, we are considered as unit time operations, then we say, that the all these operations will require 1 unit of time and we also call these unit time operations as elementary computations. So, therefore, the idea is, that you measure the total number of elementary computations, that are required for algorithm to complete its run and so this is what you define, I mean, by time complexity of an algorithm.

So, the time complexity of an algorithm would be total number of elementary computations required for an algorithm to complete its run and so this is the, then we will say, that this is the time complexity. And now, the next step would be to define what we mean by, why do, when we say, that an algorithm has polynomial time complexity? So, I will come to that.

(Refer Slide Time: 38:04)



So, as I said, polynomial time complexity of an algorithm is said to be, so algorithm is said to be of polynomial time complexity if the total number of elementary computations, that are required for the algorithm to complete its run or given output, you know, is bounded by a polynomial in the size of the problem. So, then, you can say, that the total number of elementary computations, that are required will be number, which is bounded by polynomial in the size of the problem.

For example, while discussing the Digkstra's algorithm for shortest path problem, I had shown you, that the complexity would be order n square and here of course, see even though shortest path problem has weights attached to the edges, they do not matter in the algorithm; it is only the number of nodes that determine the complexity of the algorithm.

So, it is a, and so, as I will define the word I said in the beginning, that I (( )) going to define for you, what we mean by efficient algorithms? So, this would classify to be a very efficient algorithm.

Now, why, why this focus on polynomial time algorithms? You can see by this chart, I have just given you a small chart here to show you the, compare, comparing the growth of polynomial and exponential functions.

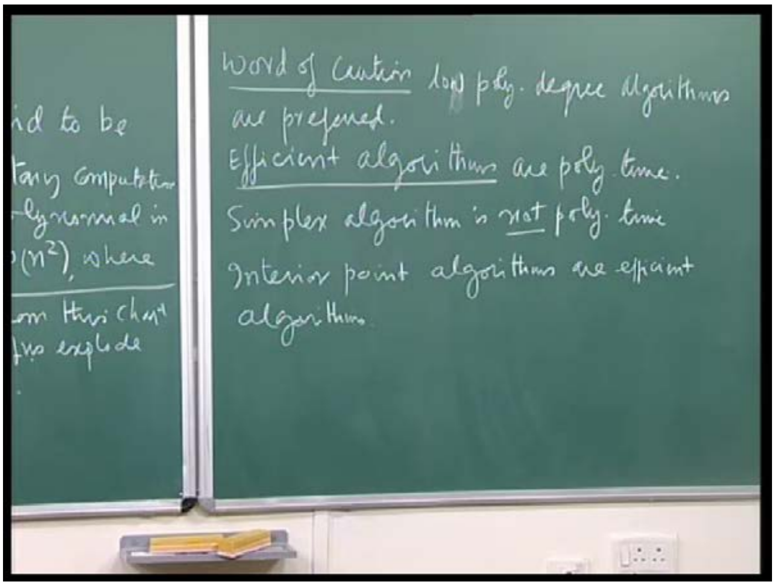
So, if you take n to be 10 for example, then n log n would be 33 and it will, this be 664 and 9966, but n, if you look, look at n cube, that again is to reasonable 1000, then 1000000 and 10 raise to 9. But if you look at 2 raise to n, which is an exponential function because n is in the exponent, then you see, for 10 itself is this number 1024, but then, here immediately the number becomes 10 raise to 39, of the order of 10 raise to 39 for n equal to 100 and for n equal to 1000, this is 10 raise to 301, which is the phenomenally big number.

And n, n factorial of course, is much bigger than 2 raise to n. So, you see, that as n increases the exponential numbers explode, whereas the growth of polynomial functions is not that, the growth is slow, but the exponential functions become out of unmanageable 10 raise to 2567 for 1000 nodes only. So, example, if you are looking at a network with 1000 nodes, you cannot think of an algorithm, which, which, which has a time complexity of the order of n square or even 2 raise to n, because this is 10 raise to

301 and only 1000, but traveling salesmen problem and so on, sometimes run and 10000 nodes because we are taking care of all the cities.

So, that is why, the emphasis on discovering algorithms, which have polynomial time complexity because you know, that even with the increase in the size of the problem, the complexity would not increase, you know, would not become unmanageable and so of course, one can say, that is it, is it enough to have polynomial time complexity because if you have the complexity, as you know, of the, the polynomial is of the order of let us say, 50 then you know, n raise to 50 is also a big number because n raise to 3 already is becoming 10 raise to 9. So the, so that is what word of caution is.

(Refer Slide Time: 41:47)



That is, that you are actually looking for low polynomial degree algorithms, but what has been experienced so far is that once you discover a polynomial time algorithm for a problem, then improvements and new this thing become possible and you can bring down the degree of the polynomial; see, this is what is remarkable.

So, therefore, once you are able to show, that the complexity of the algorithm is polynomial time, it is bounded by a polynomial in the size of the problem, then the advancements, theoretical advancements and so on, even your computational advancements have shown that degree of the polynomial can be certainly brought down.

So, therefore, and so, we will call algorithms, which are polynomial time complexity as efficient algorithms.

And then, this was found, many people really had to work hard to create data for which the simplex algorithm was shown to be not polynomial time, and this I will discuss in the next lecture to show you, that how, because otherwise the experience with the simplex algorithm has been very good, millions of problems have been solved very efficiently. When I say efficiently here, what I mean is, that the effort required was not really exponential in terms of the size of the problem, it was done in very reasonable amount of time.

But if you want to see for example, when you talk of the complexity, you talk in terms of worst case, that is, if what, what is the worst time, that will be taken by the algorithm to solve a particular problem? So, people worked hard and they did find certain instances of the linear programming problem for which they could show, that with the, with certain exit and entering rules for the simplex algorithm, the algorithm will actually go through exponential number of vertices before arriving at an optimal solution.

So, simplex algorithm is not polynomial time and I will discuss one of these examples with you and then, of course, interior point algorithms came in and they are, they have been shown to be efficient. And of course, the one I will discuss with you is a Deccan's algorithm, which actually has asymptotic convergence, what we mean by that is that. So, I will give you the outline of the interior points of algorithm, what are the issues involved here and then, discuss with you one of the basic interior point algorithm, which is known as a Deccan's algorithm or the (( )) algorithm, but here, this also is not polynomial time in that sense, because the convergence to the optimal solution is asymptotic, that means, infinite steps.

But thing is that you can always stop at a point where your, you are not willing to, you are willing to stop at a point, which is close to the optimal solution, but not actually the optimal solution and then to work around getting to the optimal vertex solution. So, then, in that case, in that sense, the algorithm would be polynomial time. So, we will continue with the description of the interior point algorithms and the example to show you, that simplex is not polynomial time.