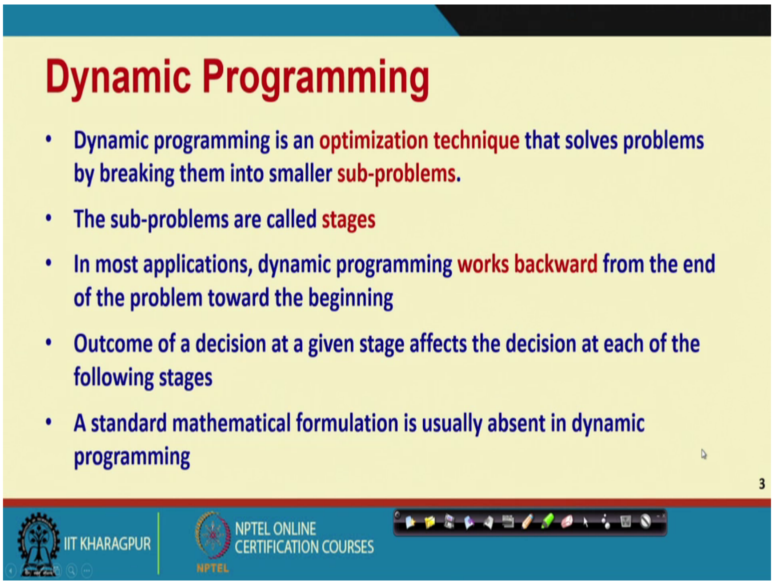
Selected Topics In Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture – 01
Dynamic Programming: Introduction

So, on the subject Selected Topics in Decision Modeling, we are now in our first lecture that is Dynamic Programming and we are in the introduction chapter, right.

(Refer Slide Time: 00:30)



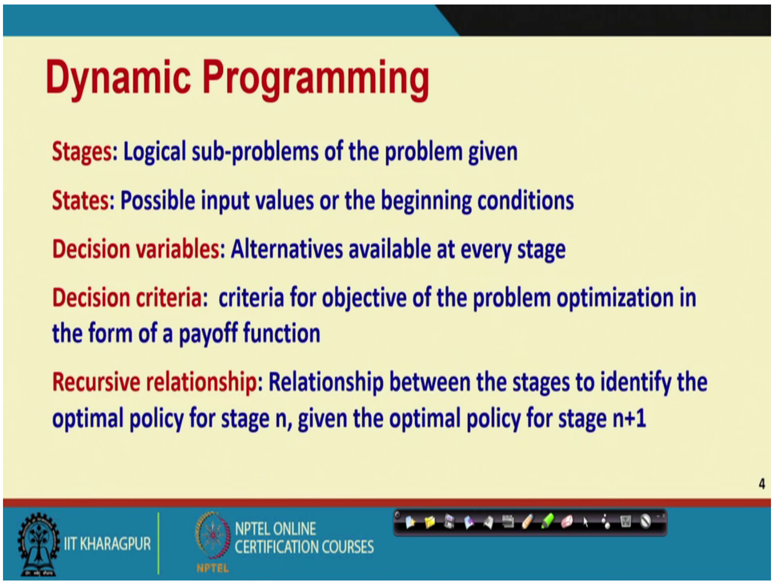
So, in introduction, the very first thing that we must know that dynamic programming is an optimization technique that solves problem by breaking the problem into a number of sub-problems, is it all right. So, you see, this is a first of all, we should know that this is a pure optimization technique; that means, it guarantees and optimal solution, is it all right. So, that is very important that you know, it is not an heuristic or in exact method, it actually gives an you know exact solution, right.

Now, when we divide these problem into sub problem, each sub problem can be called as stages, is it all right. So, we have a big problem that big problem, we divide into a number of sub problems which we call as stages, now in most applications dynamic programming works backwards from the end of the problem towards the beginning. Now, what is the end and what is beginning it depends from problem to problem. So, we shall understand this when we take up examples. Now, outcome of a decision at a given stage affects the decision at each of the following stages. So, it is like these that you know in stage 4, see in every stage, we have some states in which the decision variables can be in, is it all right and we make some decisions.

Now, let us say we are in a given stage and we find the optimal decision in that particular stage these particular optimal decision that we obtain in a given stage can be utilized in the previous stage because we are moving backwards. So, we are in let us say stage 4 and we find an optimal decision when it comes to the stage 3 optimal decision; we can make use of the stage 4 optimal decisions and you know we obtain; what is known as the stage 3 optimal decision unlike a linear programming and many such methods really speaking, we really do not have a standard mathematical formulation, is it all right, it is not like an linear programming problem where we have an optimization formulation it is not like that.

But the point has to be remember that all though there is not a standard mathematical formulation, given stage is connected to the next stage by a recursive formula. So, what is that recursive formula? How are they connected? We shall see them in due course of time.

(Refer Slide Time: 03:21)

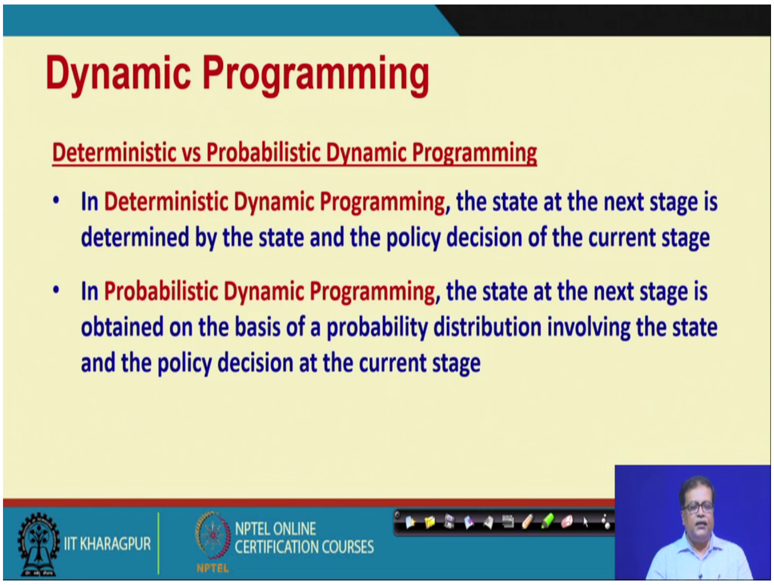


So, as I said before, there are few things of a dynamic programming the first thing are the stages which are the logical sub problems of the problem given. So, we have a problem we divide the problem into a number of sub problems and those sub problem should be logically connected to an another, is it all right that is the first thing and then possible input values the beginning conditions can be called as states, right.

So, those input values could be in multiple states in other words in every stage you know the values could be in different possible states is it not which one will be the state that will be determined by the given situation then at every stage there are certain decision alternatives which we are really obtaining, they can be called as decision variables the objective function can be really formed with this decision variables.

Then there are certain decision criteria the criteria for objective of the problem optimization in the form of a payoff function; that means, there should be a function that would really obtain the value of the objectives and the relationship between the stages to identify the optimal policy first stage n given the optimal policy for stage n plus 1; obviously, for backward dynamic programming can be called as a recursive relationship.

(Refer Slide Time: 05:05)



Now, having said that; now again there are two types of dynamic programming that we can have may be called as deterministic dynamic programming and probabilistic dynamic programming the deterministic dynamic programming is mostly; that we shall look into these are the that those kind of dynamic programming where the state at the next stage is determined by the state and the policy decision of the current stage in a deterministic manner, right. So, these will be found out with certainty and there will be no ambiguity whatsoever. So, those can be called as deterministic dynamic programming.

As a contrast, we also have what is known as the probabilistic dynamic programming where the state at the next stage is obtained on the basis of probability distribution involving the state and the policy decisions at the current stage. So, what does it mean that suppose, we are in a given stage say stage 1 and let us take a an example of an investment problem and we are in stage 1 where we make one investment. So, when we make an investment in a particular option, you see the return may not be deterministic.

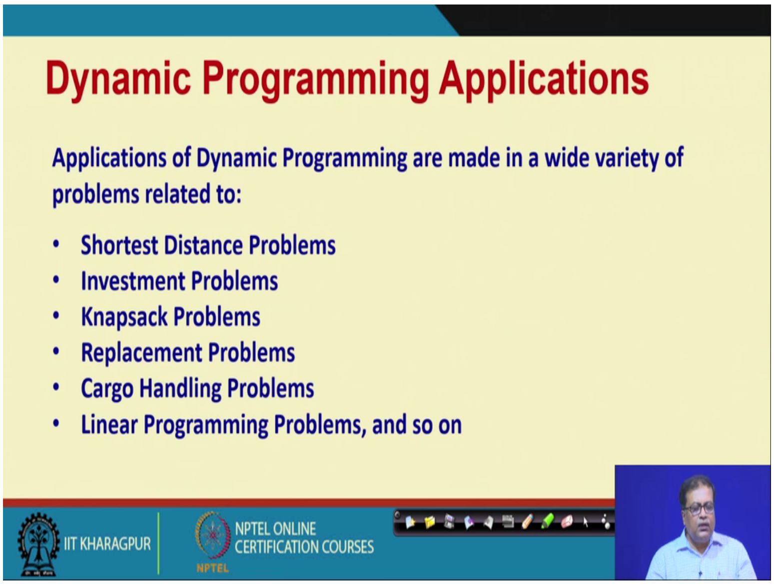
Suppose you put in a particular investment option rupees 1000, after one year your return, let us say is either no additional gain; that means, you get back your rupees 1000 or you gain 1000 rupees so; that means, rupees 2000 and let us say both are equally probable. So, what does it mean that from a state where you have 1000 rupees in your hand, let us say; the amount of investment or amount or rupee at hand is the variable that you are measuring a state. So, you had 1000 rupee in your hand. Now, after one year, either you will have 1000 rupees or you will have 2000 rupees, is it all right.

So, those are the two possible states at the next stage. Now at the next stage; obviously, your decisions will be different if you had 1000 rupees or 2000 rupees. So, it is not only that from a given state stage you go to the next stage, you know different states, but subsequently also in the future stages also your decisions and your outcomes will be different, right.

So, as you can see that these probabilistic dynamic programming will be much more involved in comparison to a deterministic dynamic programming, right. So, while in deterministic dynamic programming, if you invest rupees 1000, the return will be clearly known let us say rupees 1000; that means, from a stage 1 say with you invest rupees 1000 investment option one, then the possible state could be rupees 1000 only and not anything else, is it all right.

Obviously, if you put more money you could go to another state you know those sort of things can happen, but as long as you choose a given investment option your return will be always specific you know return cannot be anything else, right.

(Refer Slide Time: 08:51)



So, what are certain applications of the dynamic programming problem? So, there are different applications that can be made in different kinds of problems and some of thems like shortest distance problem where there are say number of cities which are connected through a network and there is a beginning city there are other cities in between the different stages and then we have the destination city and we have to find out the shortest distance from the beginning city to the ending city.

Then there are investment problems, right, what are investment problems that there are different investment options, one can choose a given investment option in a given stage, you can make certain investment you know certain amount or more in a given stage, as you put certain amount of money depending on how much amount of money you have put, you will get certain amount of return and then you choose the next investment option and etcetera. Finally, you have to choose that how you choose between the different investment options so as to get maximum return.

Then you have those knapsack problems what really happens in knapsack problems is that you know as if you have a sack and this sack has got certain capacity, let us say in terms of weight right, only a certain amount of weight can be carried by the knapsack and you different items that you have of different quantities and I mean different values and their different weights. So, how many of those different items will you be able to put in the knapsack of a certain weight. So, that you get the maximum possible value, right. So, those are called knapsack problem.

In fact, a type of knapsack problem is a project selection problem just imagine that we have different projects and if you do a given project you get a certain return. So, here return is like value and project requires an investment which is like weight. So, how will you choose your projects?

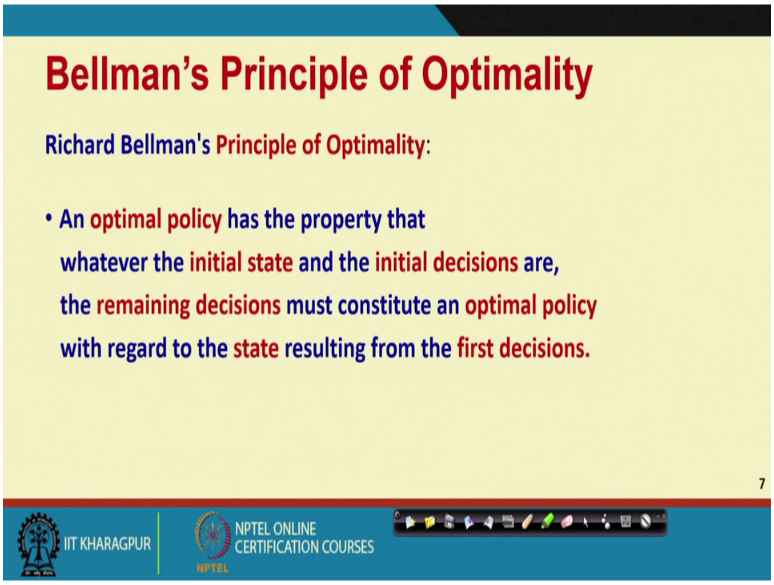
So, that you know you get maximum return of the money that you have invested in those projects then there are replacement problems, right. So, in replacement problems what really happen suppose you know you have certain equipment and the equipment has a life? So, you buy the equipment with certain amount of money and you use the equipment for certain number of years if you use n years you get certain salvage value; if you use more, you get maybe less salvage value.

So, what is the time in which you should replace your equipment is it all right and let us say if you are given total number of years within those total number of years how many times should you replace you know those equipment. So, that with minimum possible investment is it not including the salvage value consideration you can operate the machines, is it all right.

So, those are replacement problem then there are cargo handling problem is it not. So, there are let us say truck the truck can carry only a certain amount of weight; how do you fill up your cargo. So, that you know with certain number of items so that you can carry maximum value with a certain weight restriction.

So, as you can understand the cargo handling problem is also a type of knapsack problem then linear programming problems can also be solved by dynamic programming although will not discuss them in our current set of lectures, right.

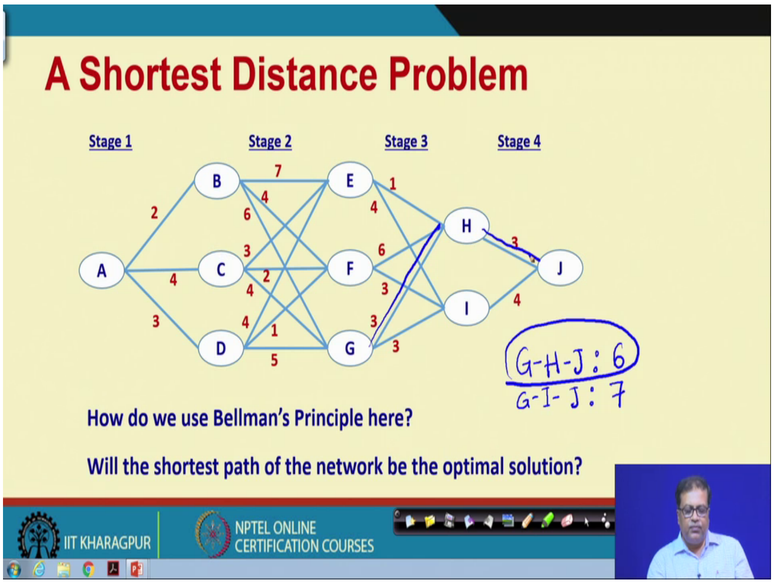
(Refer Slide Time: 13:16)



So, having said that; now come to the most important part that is known as the Bellman's principle of optimality, right, Bellman's principle of optimality is stated in this manner that an optimal policy has the property that whatever the initial state and the initial decisions are the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions is it not.

So, what it does mean? It mean that you know let say we are moving from a given state to another state. So, let us look at this problem.

(Refer Slide Time: 14:10)



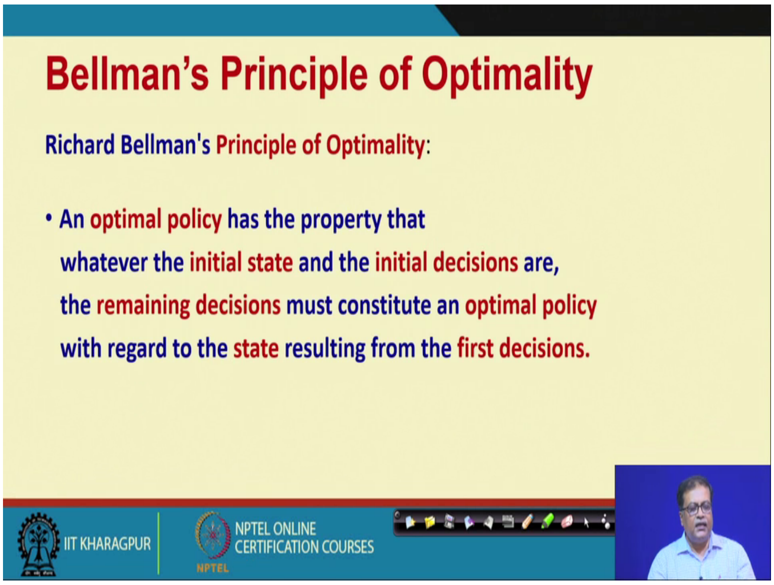
Let us take a shortest distance problem. So, in the shortest distance problem, if you see there are certain number of cities and let us say, we have a city A, from the city A, we are moving to the city J. Now you know as we move from city A to city J, there are many other cities in between like there are cities B, C, D and then there are cities E, F, G and there are cities H I and so on and so forth. Now you see from the city A, you can move to either B or C or D, right. Now from B, you can move to again E or A, for G, you can also move to E or A for G from C or D.

Now, in this particular example, you can move from any city to any other city in a realistic problem; it may not be so that from each city, there is a path to the next set up cities, they are may not be path also, is it all right. So, what Bellman's principle speaks about here supposing you are moving from A to J and at this point of time, you are in city G, right. So, where are you are in city G, right? So, you are in city G and you have to go to city H, right. So, you are in G and you are going to H. So, what is the optimal path from G to J, you see your destination city is J and you are in city G.

So, Bellman's principle once again if you if you think, you know what does it speak? It speaks whatever your initial decisions are; that means, it does not matter how you have come to G, is it all right, does not matter, let us forget how you have come to G from G to J you must follow an optimal path, is it all right. So, you know how what is that optimal path from G to J, please look at this diagram carefully and you tell me what is the optimal from G to J, you see what are some distances if you go through H, then your distance will be 3 plus 3; 6, if you go through I, then your distance will be 7.

So, you know let us write them here if you go from G to H to J, then your distance is 6 and if you go from G to I to J, then your distance is 7. So, if I compare these 2 path, then from G to J the optimal path is this one is it not. So, we can also indicate that G to H to J. So, supposing I make double line this portion. So, you can see that from G to J the optimal path is G to H to J that is 6, right. So, let us look back and then understand the what is that Bellman's principle once again right let us go back one slide and see what it says.

(Refer Slide Time: 18:16)



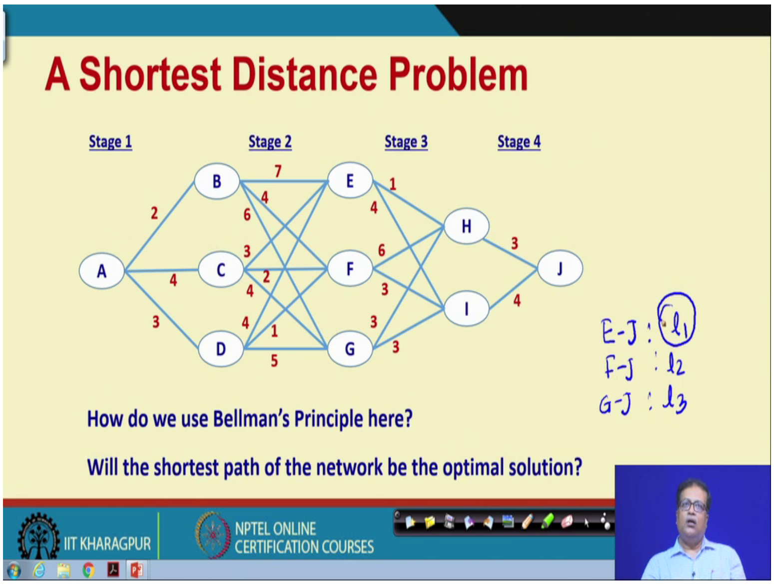
And optimal policy has the property that whatever the initial state and the initial decisions are the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions, all right

So, now see we what we found once again let us write down you see we have written down these that G to H to J was our optimal decision, is it and what Bellman's principle tells once again that whatever our initial decisions are you see if I have to go to J, either I have to go from E or through F or through G at this particular level, is it all right.

Now, assuming that we have come to G; whatever way we might have come to G the remaining decisions must be optimal. So, what is that remaining decision the remaining decision comes out to be G to H to J which is 6 and which is optimal; that means, if we are at G, we should now choose the next you know next city has H and go there. So, that I can complete the remaining journey in the optimal way by a distance of 6, is it all right.

So, this is what it says. So, extending that if we would have been in F again I can find the optimal decision, if I would have been in E again I could find out the optimal path from E to J, is it ok. So, just look assuming.

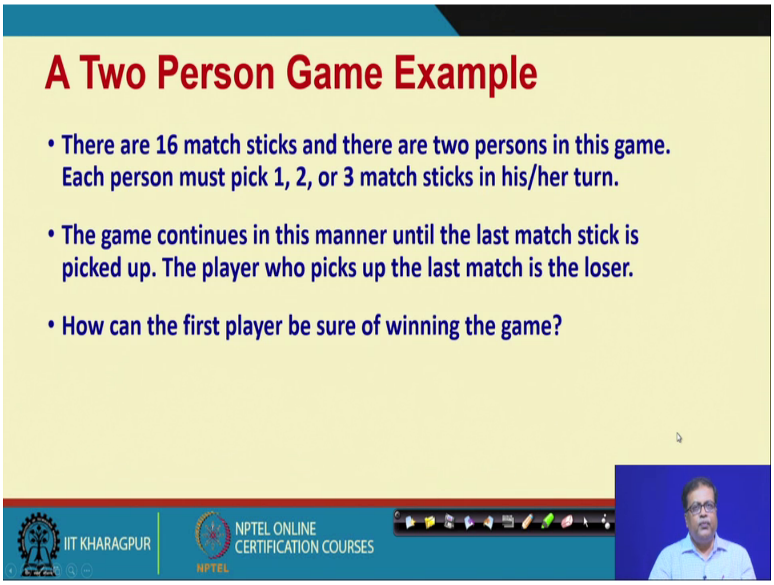
(Refer Slide Time: 20:18)



I know E to J, I know F to J and I know G to J supposing these are l 1, l 2 and l 3, right. So, assuming that they are l 1, l 2 and l 3 and just assume for the time being; suppose, l 1 is the lowest or the shortest; so, if l 1 is the shortest, then you know you understand that in order to move from this stage to the destination state, J we should move from E and not G and how do you use this fact A further on you know ah; that means, the optimal path should be from E and that path we shall use.

So, we can have that and we go back to the previous set of cities use this fact and you know do further decisions. So, this is how really the dynamic programming works. So, I did not really go into the details of it, I just tried to tell you the how the whole thing is happening right as a you know general example.

(Refer Slide Time: 21:46)



So, having said that let us look at another problem; you know at two person game a simple two person game; example supposing there are 16 match sticks and there are 2 persons in this game, each person must pick 1, 2 or 3 match sticks in his or her turn the game continues in this manner until the last match stick is picked up.

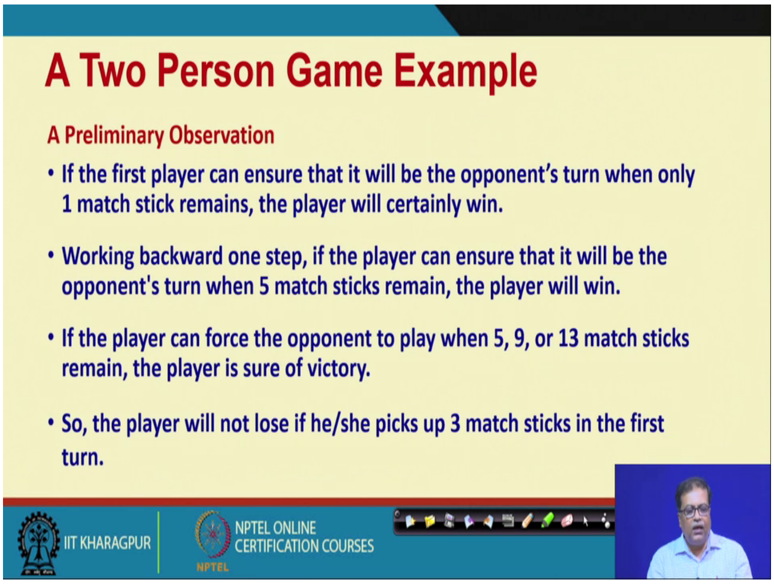
So, the player who picks up the last match is the loser; how can the first player be sure of winning the game. So, I hope you understood the game; there are 16 matchsticks and now it is your turn, right, your friend is also playing and you want to win; who loses whoever picks the last match stick.

So, what should be your goal to win in this game to create a situation that your friend picks up the last match stick; so, if your friend picks up the last match stick, then you win. Now, how do you force your friend to pick up the last match stick? So, look here whatever your initial decisions are if only one match stick, suppose, there are 4 match sticks remaining just imagine there are 4 match sticks remaining.

How many match sticks will you pick up. Now, you will you pick up 3, is it right. So, that only one match stick is remaining. Now, your friend has to pick up because in your turn, you have to pick up something either 1, 2 or 3. So, what your friend can do he has to pick up that last match stick and lose; is it not. So, you can understand that if 4 match sticks are there you can win the game, is it all right, before your turn; before your turn. So, that you know you can, now pick up something like 3 match sticks and win, but suppose 3 match sticks are remaining, you can pick up 2, if 2 match sticks are remaining, you can pick up 1, is it all right.

But if 5 match sticks are remaining, what will you do then? Right, you know it will be slightly difficult to decide if 5 match sticks remaining; what should be your strategy? Your strategy should we pick up one match stick so, that 4 match sticks remaining. Now, your friend you know, but then even if you we take one match stick, your friend will be now in the driver's position, your friend will then pick up 3 match sticks and then you lose, is it all right. So, I hope you understood the game.

(Refer Slide Time: 24:43)

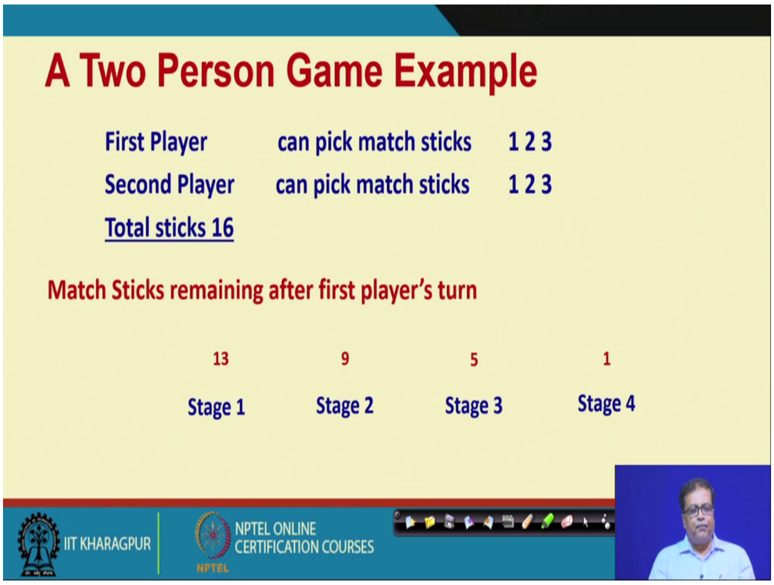


So, have you understood the game you know. So, if the first player can ensure that it will be the opponents turn when only one match stick remains the player will certainly win.

Working backwards; one step if the player can ensure that it will be the opponents turned when 5 match sticks remain, the player will be win. So, if the player can force the opponents to play when 5 nine or thirteen match sticks remain the player is sure of victory. So, the player will not lose if he or she picks up 3 match sticks in the first turn.

So, look here if you can force your friend to lose if you live only one match stick is it all right. Now, 4 is the key here supposing, if your opponent player is playing and he has only 5 match stick remaining, is it all right, if you picks up 1, 4 will remain, then you pick up 2, you pick up 3 and then your friend will lose.

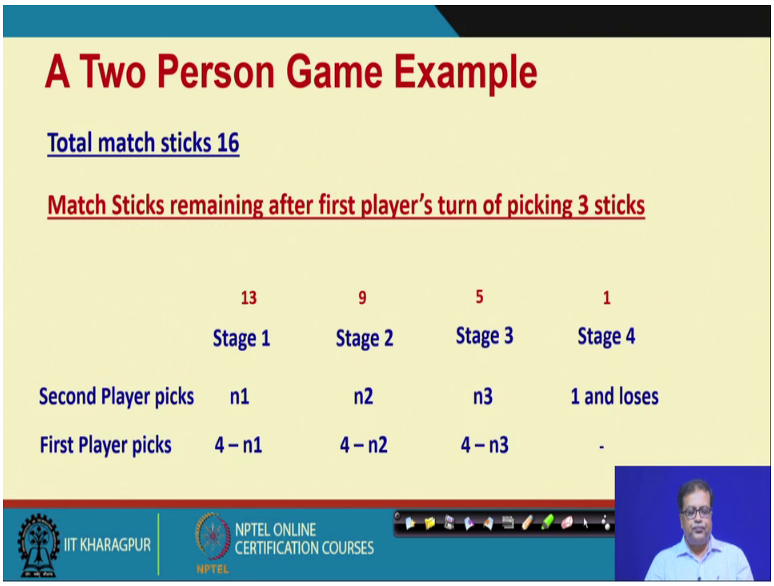
(Refer Slide Time: 25:57)



So, you can understand this that how this is really happening. So, first player can pick match sticks 1, 2, 3, second player can pick match sticks 1, 2, 3, total 6; 16.

So, in stage 4 if 1 is remaining after your turn, then you win, in stage 2, if 5 is remaining, then you win, in stage 3 if 9, then you win, in stage work if 13 that you win, right and since there are 16 match sticks; that means, the beginning you pick up 3 and then you win your game.

(Refer Slide Time: 26:29)

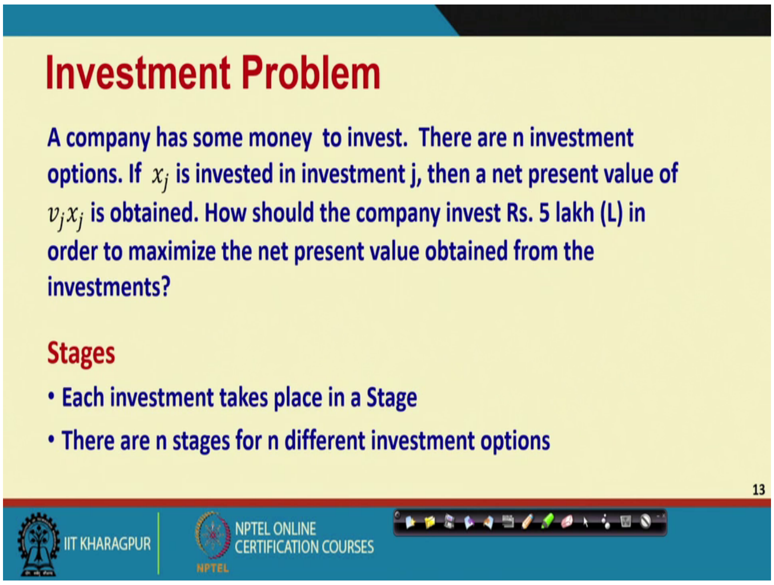


So, essentially so, you see the second player and then how it should go that at any stage you know supposing you pick 3 sticks there are total 16. So, now, thirteen remaining if the second player picks n 1 you pick 4 minus n 1, is it all right, then how many will remain; 9.

So, if you picks n 2, you pick 4 minus n 2 5, n 3 4 minus n 3 1, he takes 1 and loses, is it all right. So, you see what we have done from stage 4, we subsequently you know at the optimal value for before second players turn is 1, we go back because 4 is a magic number here. So, we go back you know 4 more in every subsequent stage and create a number and we force that number before the second player play and play with a strategy we are sure to win, is all right.

So, what has happened here? Stage 1 stage wise, we have really employed; what is known as the dynamic programming principle.

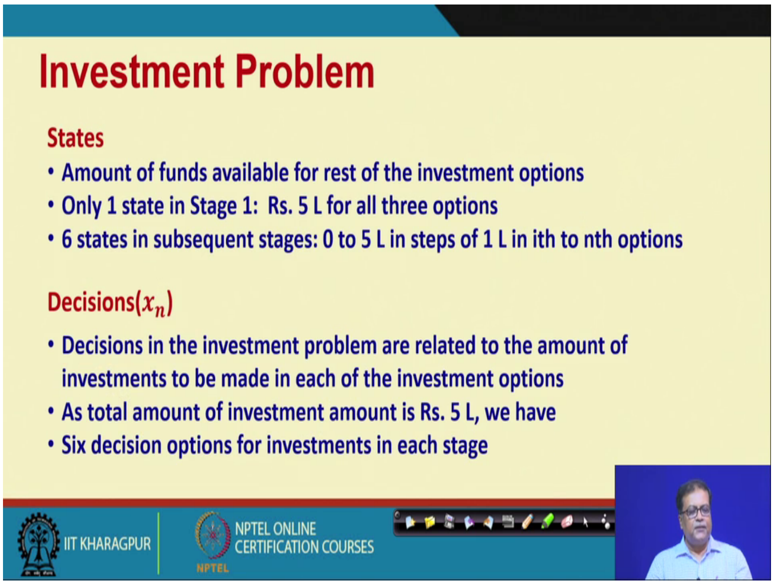
(Refer Slide Time: 27:52)



Now, let us take an investment problem and in a some more detail and try to understand this from the dynamic programming point of view, a company has some money to invest, there are n investment options, if x j is invested in investment j, then a net present value of v j x j is obtained, how should the company invest rupees 5 lakhs, call L; let say in order to maximize the net present value obtained from the investments.

So, what are the stages in each investment takes place in a stage and there are n stages for n different investment options, but point is at a given investment options you can invest you may not invest, is it all right or you can invest more than 1 lakh in stages of 1 lakh. So, it is it is could be that in the first investment option you can put your entire 5 lakhs or nothing or anything between 0 to 5, is it all right. So, if that is so, then the stages are that there are n stages for each investment options one stage.

(Refer Slide Time: 29:06)



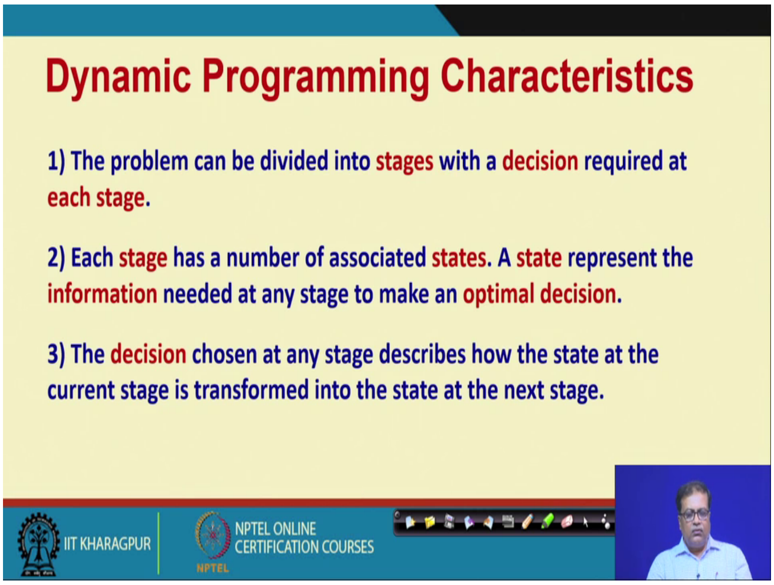
What are the states? The states here could be the amount of fund available for rest of the investment options see, let us say there are 3 investment options.

So, initially you have 5 lakhs available, right. So, assuming you put x 1 amount in the investment option 1, then 5 minus x 1 is available in the beginning of stage 2, right assuming x 1 equal to 0, then entire 5 lakhs are available. So, possible states could be you have either 0, 1, 2, 3, 4, 5; all of them available for the subsequent stages, is it all right.

So, if x 2 is also 0, then again at the third stage all 0 to 5 options are available, but if x 1 equal to 1, then you have 0 to 4 available, is it all right. So, those are states and decisions the decisions are amount of investment to be made in each of the investment options, right.

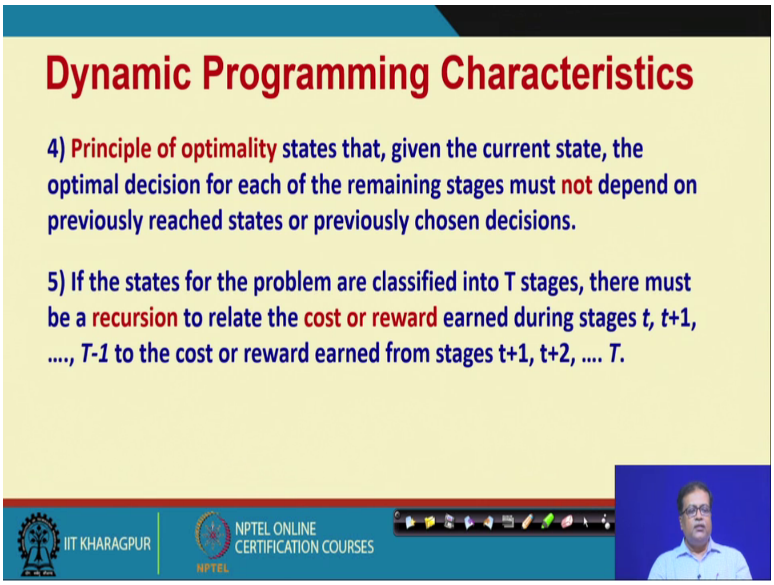
So, as total amount of investment amount is 5 lakhs, we have 6 decision options for investments in each stage that is 0 to 5, right.

(Refer Slide Time: 30:25)



So, now dynamic programming problems can be divided into stages with the decision required at each stage. So, each stage has the number of associated states a state represents the information needed at any stage to make an optimal decision the decision chosen at any stage describes how the state at the current stage is transformed into the state at the next stage right.

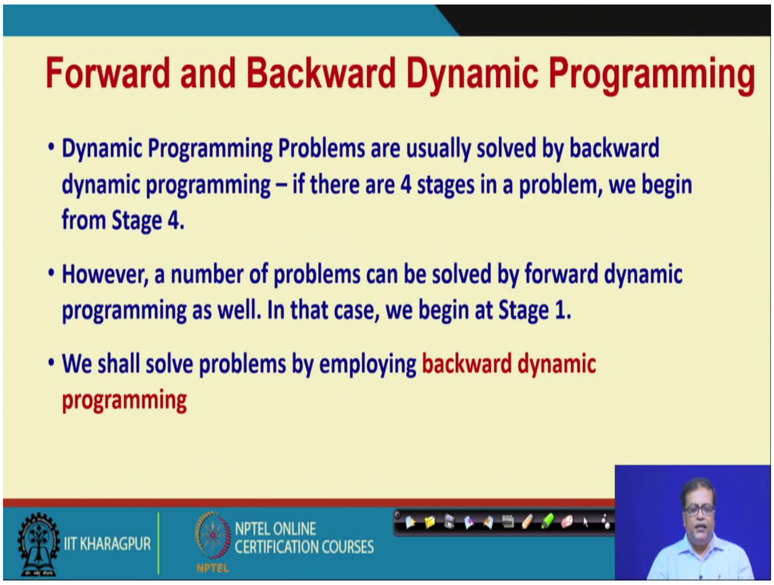
(Refer Slide Time: 30:52)



Then the principle of optimality states that given the current state the optimal decision for each of the remaining stages must not depend on previously reached states or previously chosen decisions.

If the states for the problems are classified into t stages there must be a recursion to relate the cost or reward earned during stages t, t plus 1, etcetera to the cost or reward earned from stages t plus 1, t plus 2, etcetera. So, these we have already seen in our previous example when we discuss the distance problem, right.

(Refer Slide Time: 31:32)



And then finally, the dynamic programming problems are usually solved by backward dynamic programming. So, if there are 4 stages, we begin with stage four; however, a number of problems can be solved by forward dynamic programming as well in that case we begin at stage 1, but in this particular set of lectures, we can solve problems by employing backward dynamic programming, right.

So, thank you very much. In our next lecture, we shall take up the stage coach problem which is a shortest distance problem.