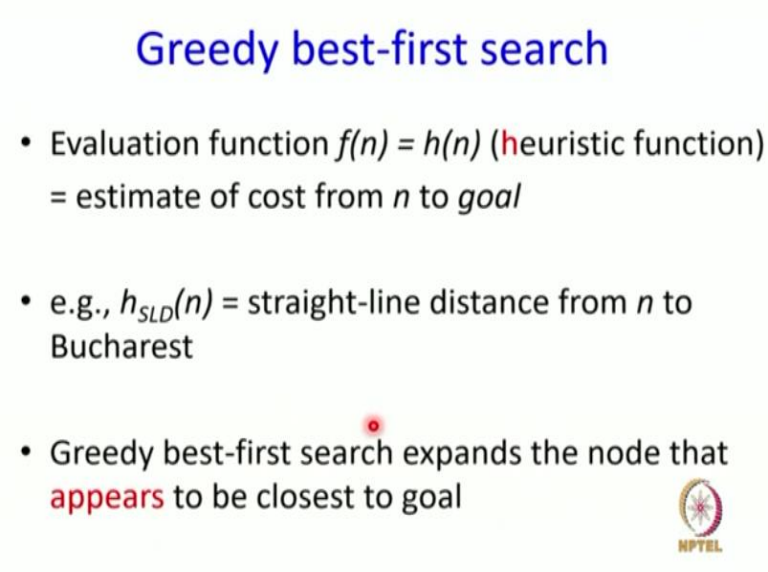
**Artificial Intelligence**
**Prof. Mausam**
**Department of Computer Science and Engineering**
**Indian Institute of Technology – Delhi**

**Lecture 16**
**Informed Search: Greedy Best First Search and A* Search**

I want to make sure that you are comfortable with all three of these, because these are important. F of n says which is the best node overall cumulatively, g of n says which is the closer node from the start to right and h of n says how far is the goal from me. Now if you are using or if you are devising an algorithm, a search algorithm, which somehow used the h function. What was my intuition? I want to go to a node, which appears closer to the goal.

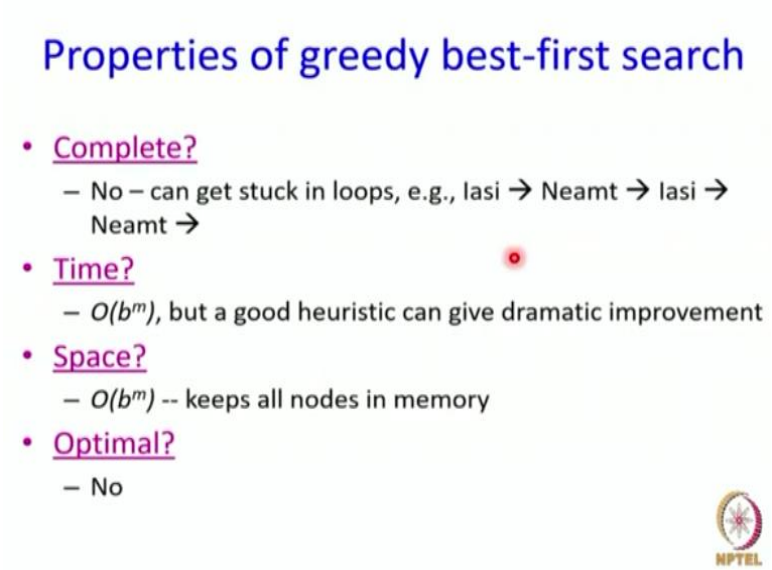
So if you wanted to change you best first search algorithm to formulate this inside, you will say f of n is equal to h of n, everybody with me. I want to make sure everybody is with me. So let us stop at this point.

**(Refer Slide Time: 01:22)**



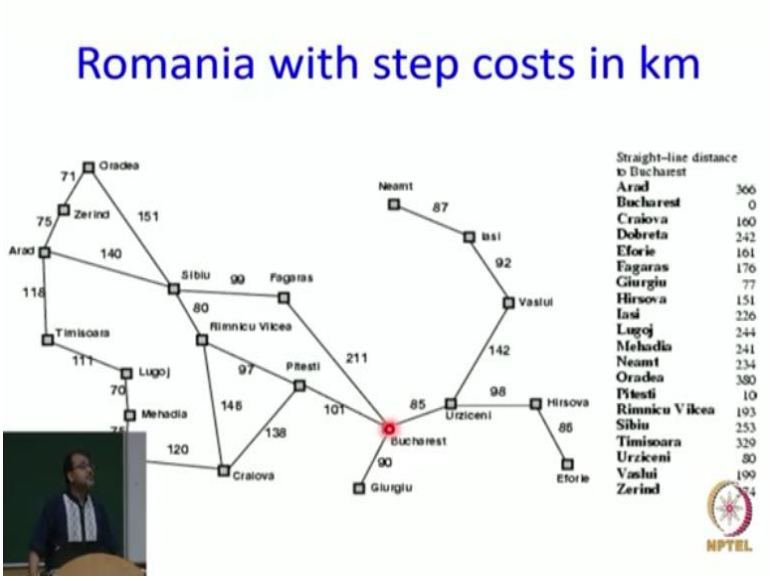

This algorithm is called the greedy best-first search. It says evaluation function f of n is equal to h of n. The node I am going to expand next is the node, which gives me the hope that it is going to reach to a goal with the lowest cost. Its h function is the lowest and for example, we just did one kind of h function, it is a straight line distance from n to the goal, Bucharest. Everybody with me? Any questions?

**(Refer Slide Time: 02:01)**




Now comes the important question, is it a good algorithm? To understand this, let us go back to our map of Romania.

**(Refer Slide Time: 02:18)**

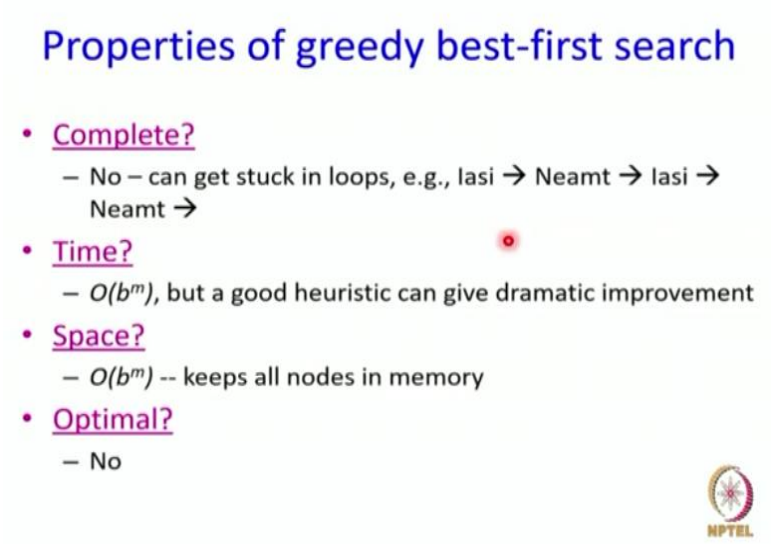



And I am going to give you a slightly specific example, different example from Arad to Bucharest, but the one that makes the point. So suppose my goal is to go from this city called Iasi to this city called Oradea. This is my goal. I want to go from Iasi to Oradea and let us say that I use the greedy best-first search algorithm.

So let us run it. So from Iasi, what are the cities I can go to? How many cities can I go to? Two, one of them is Vaslui and one of them is Neamt. Now which of these has a better heuristic function with respect to Oradea? Neamt. So what is Iasi going to say, let me go to Neamt, so far so good. From Neamt, how many cities can I go to? One Iasi, which of these cities look closer? Iasi. So I will go to Iasi and from Iasi again I can go to which cities? Neamt and Vaslui.

Which one is closer? Neamt and so basically I will keep shuttling between Iasi and Neamt and Iasi and Neamt and Iasi and Neamt and I will never explore anything else in the fringe. This is on the T saturation. The question is I have been not doing duplicate detection on the T saturation.

**(Refer Slide Time: 04:13)**




So this tells me that at least for tree search, greedy best-first search is not complete. Now you can ask what about the graph search version, obviously?

And there greedy best-first search is actually complete. So they are differing properties and finally in the exam, I might ask you a question about some algorithm in the setting where it is graph search or tree search with duplicate or partial duplicate detection and so and so. So I can ask you all kinds of questions. You have to think about properties of the algorithms along various dimensions. Now time, let us think about time.

Do you have any guess what is the time going to be? Notice that the time will be heavily dependent on the heuristic function, because this is the additional information, I have given you. I have given you additional heuristic function. I can give you a terrible heuristic function or I can give you a really, really heuristic function. In fact, I can give you the optimal heuristic function. What is the optimal heuristic function? The actual optimal cost from that node to the goal.

I can give you the optimal. We can call it h star. The optimal is h star or I can give you a heuristic function, which completely confuses you. In fact, it throws you off. It takes you in the wrong direction, like somebody defined an h, so that I get confused, my intuitions get completely confused. That will be a Hollywood story, kind of a thing. Somebody confuses somebody, and makes them do something exactly opposite of what was optimal.

So in the worst case, if my heuristic function is really crappy, in the worst case, what is going to happen? I will end up exploring all the nodes, exactly and in that setting, my time would end up being order b to the power n. So therefore, from now on, we will not worry that much about time complexity per se, because when I give you a bad heuristic function, all bets are off and of course, theoreticians then study that what was error bound in the heuristic function.

And based on that, can I do better bounds then b to the power n, but for our practical purposes, we are just going to say that look, a good heuristic can give a dramatic improvement, but in the worst case, it will be terrible, both in time and space actually. Believe it or not, because my fringe can become too large and so I may end up keeping all the nodes or most of the nodes in them. So my space and time bets off.

Of course, if it is not complete, then likely it is also not optimum. So therefore, greedy best-first search does not have grade properties. Not only does it have time and space, which is going to be hard, because we are dependent on the heuristic function, but it is not even complete. It gets stuck in loops and that is not very good. We do not want an algorithm that gets stuck in loops. So we need to do something about this. So intuitively, what is going along?

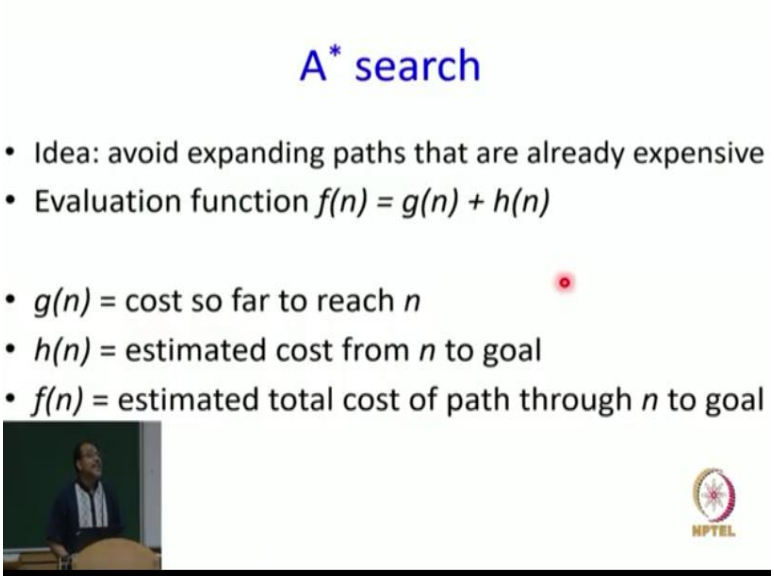
Yes, we are repeatedly visiting some states and that is going along. I do agree with that, but in terms of cost, what is going around? So if you think about it, initially when you expand at Neamt, what was its g function? G function is the cost to itself, g function was 87 and then you had some straight line distance to the goal and Vaslui had a g function of 92, but we did not even take care of g function at all. We just said oh, Neamt looks good, because it is closer to the goal.

So far, so good, but then when I came back from Neamt to Iasi, for this Iasi, the g function has become 174, 87 and 87. This g function has become 174. However, the g function to Vaslui is still 92, because it is from the first Iasi not the second Iasi recorded and as we keep going from Iasi to Neamt to Iasi to Neamt to Iasi, we are increasing the cost of the current path, but we are ignoring the cost to the current path and only looking at myopically to the extra cost that we have to give to the goal, which is what we are somehow taking the minimum over.

That is not ideal. See what is fundamentally our objective? Our objective is to go from the start state to the goal, not from this state to the goal. This state is somewhere in the middle, but our objective is to go from the start state to the goal and as we are taking the minimum, we should not myopically only take the minimum with respect to how much cost we are going to pay in the future, but also take minimum over the cost that we have paid so far.

Somehow take the minimum over this cumulative cost, not just the future cost, does that make sense? So what would a suggestion, for the change in the f function that operationalizes this intuition. Somebody says, alpha g of n plus beta h of n and I have got to simplify this for now and we will come back to your suggestion later. So I am going to simplify it to simply g of n plus h of n.

**(Refer Slide Time: 10:31)**




And this algorithm that comes out is called the A star algorithm. So the idea is avoid expanding paths that are already too expensive, even though they look closer to the goal and that we achieve by saying not only look for the future cost, but add the past costs of that. So g of n is the cost so far, h of n is the estimated cost to the goal and so therefore f of n, which is equal to g of n plus h of n and A star should be thought of. What is the intuition?

It is the estimated cost of the path from start state to the goal through n. See if you can get that intuition. It is not only looking at h, which is greedy best-first search. It is not only looking at g, which is uniform cost search, it is same. Let us find that node, whose estimated cost from start to the goal through it is the best, everybody with me? Any confusions? This is an important point in the algorithm. We can stop here. Thank you.