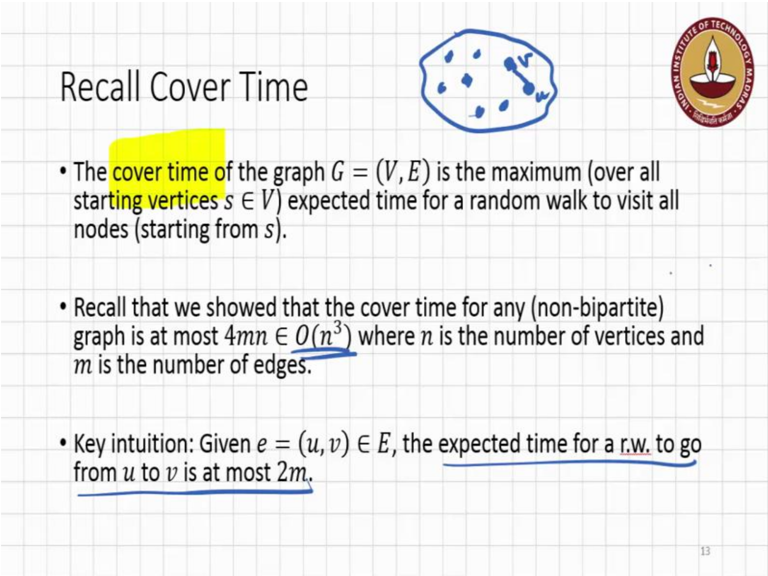
**Algorithms for Big Data**
**Prof. John Ebenezer Augustine**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Madras**

**Lecture – 34**
**Recall Cover Time**

(Refer Slide Time: 00:14)



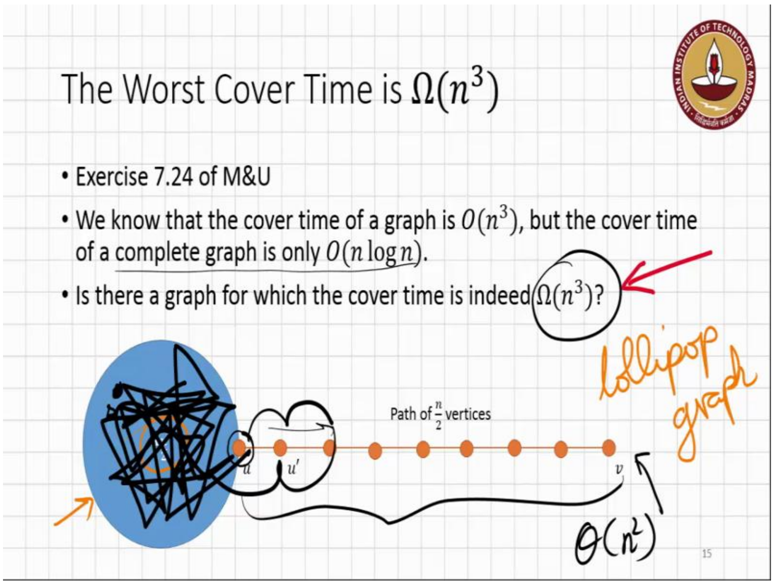

So, now that we have looked at this doubly stochastic matrices, we looked at the stationary distribution of two different linearly structure random walks, let us generalise now to arbitrary graphs. Let us recall one thing, that we have already looked at the, in the lectures on random walks.

First of all we find something called the cover time and the cover time is basically the maximum over all this possible starting vertices s. So, that is the set of all, V is the set of all vertices, the expected time for a random walk to visit all nodes starting from s.

So, now if you have a graph and this is 1 candidate, 1 vertex start of random walk from that vertex and you ask what is the expected time to cover or see or touch all other vertices so, but that would be just for that 1 vertex. Now, what you will have to do is consider, that expected time for all vertices and then take the maximum over all the vertices and that would be the cover time. And in the lecture we actually showed, that the cover time for a non-bipartite graph is at most 4mn, which is nothing but O of n cubed where n is the number of vertices and m is the number of edges.

And if you recall, the key intuition for the proof was to look at this graph and so, here if you look at an edge in this graph and let us say this is vertex u and this is vertex v, the and now let us say, you start a random walk from u. What we first showed was that the expected time for random walk to go from u to v is at most 2m and then to complete the proof what we did was, we considered a spanning tree of this graph and then we argued that we could cover the spanning tree in at most 4mn time steps. So, that was I am used to if you recall (Refer Time: 03:01) expectation to do that.

(Refer Slide Time: 03:05)




Now, the interesting question and this is the question based on exercise 7.24 of the Mitzenmacher-Upfal textbook. So, we know, that the upper bound on the expected running cover time is O of n cubed and if you look at, for example, the complete graph on the, complete graph of n vertices, the cover time there is just O of n log n. So, some graphs have very high cover time, however, very fast cover time. And this, if you recall, you may want to prove, that this is in fact, correct and I just want to remind you that this is somehow connected to the coupon collectors problem.

So, with that being the case, the obvious question now is O of n cubed upper bound on the cover time really tight. In other words, is their some bad graph for which the cover time is indeed big omega of n cubed and is as it turns out it is in fact tight and interestingly the graph that displays such a large cover time has this very nice structures called the lollipop graph. The lollipop graph has two parts to it. There is the complete graph part, which is especially in about half the vertices are all completely connected to each other. So, this is the K n by 2. And then, we have the handle of the lollipop, which is another n by 2 vertices, but now connected linearly. Now, let us try to think about why this is in fact, a case.

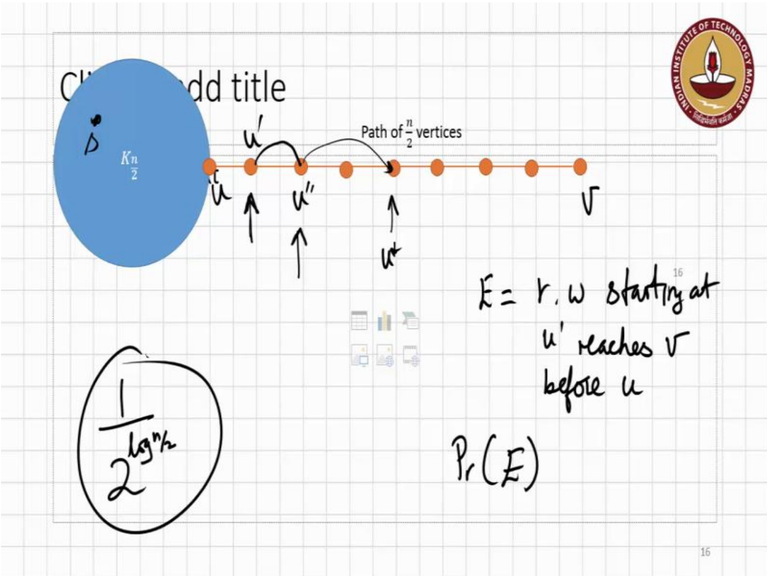
Now, let us for the moment ignore the complete graph part of the lollipop and one way to do that is just say, that let us say, you start a random walk at this vertex v and you ask, okay, how long does that random walk take to reach u and this is a question that we have already discussed in the lecture and that recall will take theta of n squared, on expectation of course. Now, why does the whole graph, so for example, if you start at v in theta of n squared time, it will reach u and once it reaches u, we have already seen that the complete graph has a cover time of just O of n log n. So, once it reaches u, the random walk will fall into this complete graph and the entire graph will be covered in an extra O of n log n time.

So, so the question therefore is, why are we making this claim? And the reason for that is, is, that you know, in the definition of the cover time you have to not just consider the start any one vertex as the starting point. You should consider every vertex as a starting point and then compute what is the worst cover time. Let us actually instead of starting at v, let us start at some vertex inside the complete graph part. Let this be our starting vertex s. Now, what will happen is that this, this random walk will start walking around this graph and it, at some point it will have to come to u and then somehow make its way all the way to v.

What makes this challenging is the following. So, if it comes to u, then there is a actually good chance it will fall back into this complete graph part. So, as long as it is inside this complete graph part, it is going to be making random walks steps inside of this region and only with probability 1 over n, it is going to come to this vertex n and then, from this vertex even if it were to make a few, say a transition like this, then with some good probability it will come back here and then it could still fall into this complete graph part. So, the random walk in some sense can easily get stuck in this complete graph part and then it takes a lot of effort for it to leave this complete graph part and then move all the way to v.

Even if it were to make a few transitions, it could with some reasonable probability and you can think about it intuitively and I also encourage you to write a small program to actually verify what I am claiming here, it could then easily fall back in here. So, really, to work its way all the way to v is takes some effort and that is where this becoming of n cubed comes in. So, let me give you a little bit more intuition as to why this is actually n cube.

(Refer Slide Time: 08:50)




Now as just as soon that we are starting at s, well actually, let us, let us, let us not do that. Let us, in fact, as (Refer Time: 09:12) consider what happens when we say or here, suppose we started at s and we have reached over here, now we want to know what is the probability that this random walk that has reached this vertex. Let us call this vertex u prime, let us call this is u; this is v, what is the probability that the random walk will reach v before it reaches u? So, the event we are interested in is, so the random walk starting at u prime reaches v before reaching the vertex u. So, and we are interested in understanding this probability.

Now, this can be actually understood reasonably easily. It takes a little bit of effort to see, I mean not effort, but just a little bit thought to see this intuition. Now, from u you are equally likely to go from u to u. Let us consider this as u double prime. From u, let us see whether we will reach u first or u double prime first and that is a symmetric random walk. So, these are two symmetric events, so their probabilities are equal and so, with probability half the random walk will reach u prime.

Now, that you have reached u prime, now ask the question, will we reach u first or will we reach this vertex, say, u star? And this, again these are two symmetric events, the event of reaching u or the event of reaching u prime, u star. So, that again the probability that you go here before you go to u is again another half. So, so what, what you can see is that with probability half you will reach u double prime before u and after having reached u double prime with another probability half you would reach u star before you reach u and so on and so forth.

If you think a little bit and work out the mathematics, you will realise, that with probability 1 by 2 raised to the something like log n by 2, you will reach v before you will reach u and working through that should help you reach this goal of showing that the cover time for this graph is in fact, becoming of n cube.