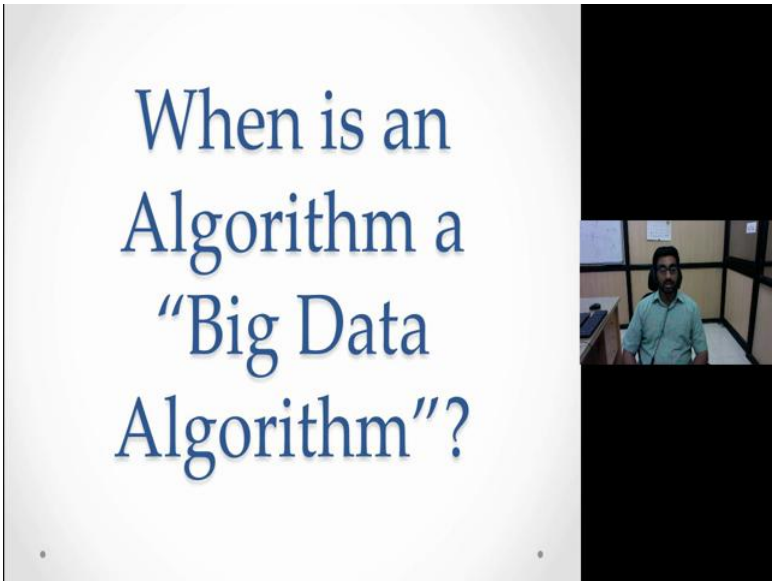
Algorithms for Big Data
Prof. John Ebenezer Augustine
Department of Computer Science and Engineering
Indian Institute of technology, Madras

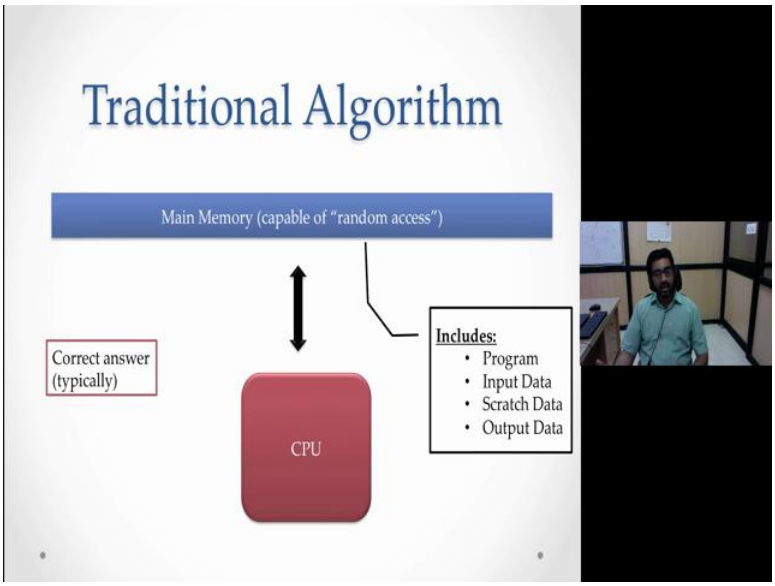
Lecture – 14
Intro to Big Data Algorithms

(Refer Slide Time: 00:15)



Hello everybody. So, now we have finished 2 weeks in this course on Algorithms of Big Data. And it is now time for us to take stop a little bit. We learned a little bit about probability theory, we looked at thinks Chernoff Bound things like, but why exactly we are learning these notions? The key to this lies in our understanding of what we mean by algorithms for big data. When is an algorithm a big data algorithm? Just to understand this let us context this with what we know to be traditional algorithms.

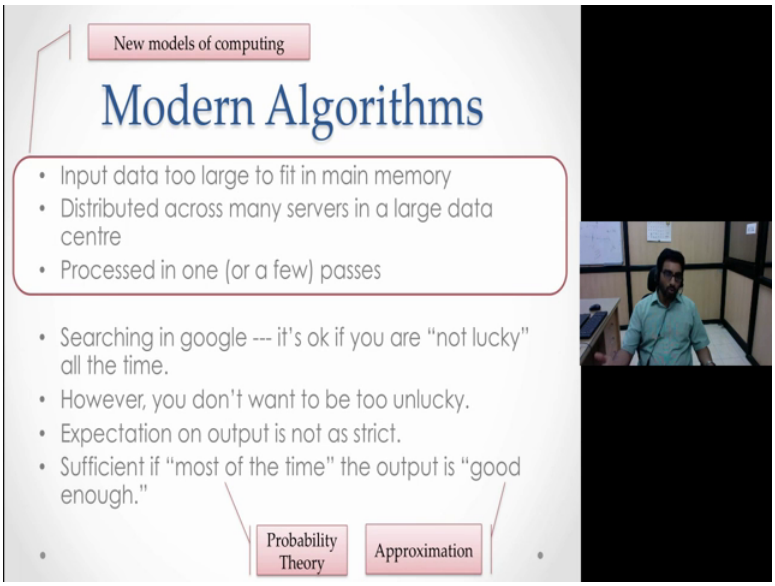
(Refer Slide Time: 00:58)



What is a traditional algorithm? We typically think of CPU that is executing an algorithm and that CPU as access to some data that is usually stored in the main memory. So, the main memory can be accessed at random locations, and the main memory includes the program that executing on the CPU the input data, the scratch information that gets generated as the algorithm is executed, and the output data everything is included in the main memory. And more over typically when you run a traditional algorithm, you accept the answer to be correct.

So, let us look at an example for example, if you want to compute the minimum spanning tree of a graph, then you except the algorithm to actually generate a minimum spanning tree. The same can be said of various other algorithms in all the over the last 50 years, we have a very nice rich set of algorithms there are been developed and over the course of time, some of the requirements have changed and there is still lot of need for traditional algorithms. So, it is not like the traditional algorithms or no longer useful, but as computer science and the applications and the world around as changes.

(Refer Slide Time: 02:28)



Ah there are new and interesting algorithmic questions there are being asked and this is inspired by various applications, that we broadly called big data applications.

So, what are these big data applications and what are these algorithms there are used these big data applications? What are the characteristic that we encounter well the input data is typical too large to fit into the main memory. So, the traditional view that the data fit is into the main memory is no longer necessarily true, and it is not just one computer that and handles a data typically it is a large data centre that processors a data, and because it is a large data centre the multiple servers and the data is actually broken into sub parts and stored in multiple servers they have to interact with each other to solve problems.

Finally, the data is often not allowed, we cannot access the data in traditional random access type of fashion. You would you have some rules by which you can access the data. So, as I mentioned the data could be stored in a distributed setting. It could be stored in a in a cloud and the only access you have is through some queries, or as an example here may be the data is stored a manner that you can only access the data as a single pass through the data. So, maybe you get an access to the graph a large graph as in some arbitrary fashion where you get one edge at a time, and you have to see this adjust one edge at a time and you have to answer something about graph. So, that is one more example of way in which you can access the data.

All of this implies that the traditional model of algorithm is, algorithm sign is no longer the only model that matters. That could be there could be a variety of ways in which we can define models of computing based on the specific architecture of the system specific application at hand so on and so forth.

And of course, we recall the traditional algorithms we require the answer to be always correct at least typically that is our expectation, but let think of canonical modern applications searching in Google, of course; we all familiar with this I feel lucky, but in that Google as. So, if we type something very prominent. So, for example, if you type in Narendra Modi, clearly the first link that you will get will be typically what you except this is a very prominent search term, but if you are searching for something less prominent. Then it is to be a little unlucky you do not need, you may perhaps you do not require it to be the first outcomes of your search, but let us say if is within the first page you are usually happy.

So, another words your expectation is no longer as secrete as the 4 traditional algorithms. And it is simple way to capture this is the face it is a sufficient if most of the time the output is good enough. So, most of the time kind of refers to sort of a probability theory type of thing where you wanted to be correct with some good probability most of the time, and you do not necessarily meet the output to be correct in the in the absolute sense, but it is wanted to be good enough and good approximation.

In this search engine context you want the outcome to be good enough in the sense that you thus the actual website that you are after, it is good enough if there are appears within the top say 5 hit. More over it is not like every time is search for something is always within the top 5, let us say 99 percent of the time you search for something and you what you are looking for us in the top 5 or in the first page that is great, and Google is an typically follow the sort of paradigm where most of the time 99 percent of the time this the website that you are looking for is within the within in the first page of course, the style when it is not in first page and it is hence; that you know, but that something you willing to leave and that is the essence of big data algorithms you know.

This a lot of therefore, probability theory and approximation ideas that need to be incorporated into the algorithm, but it is not like therefore, just because the user expectation as a little bit less it does not mean that you can start to I mean complete lose

control over the quality of the output, you still need to be careful to prove that with very good probability you get a very good answer. This is where the theory that we are studying the probability theory and the various algorithmic models will be helpful in designing the right type of algorithm. So, without further ado let us get started on some new ideas models and algorithm for big data.