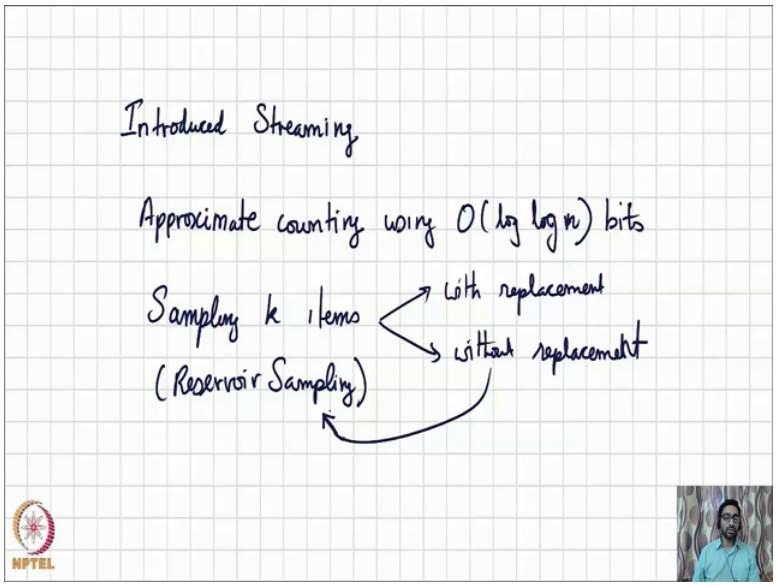
Algorithms for Big Data
Prof. John Ebenezer Augustine
Department of Computer Science and Engineering
Indian Institute of technology, Madras

Lecture – 22
Approximate Median

In this segment, we are going to look at Approximate Median.

(Refer Slide Time: 00:16)



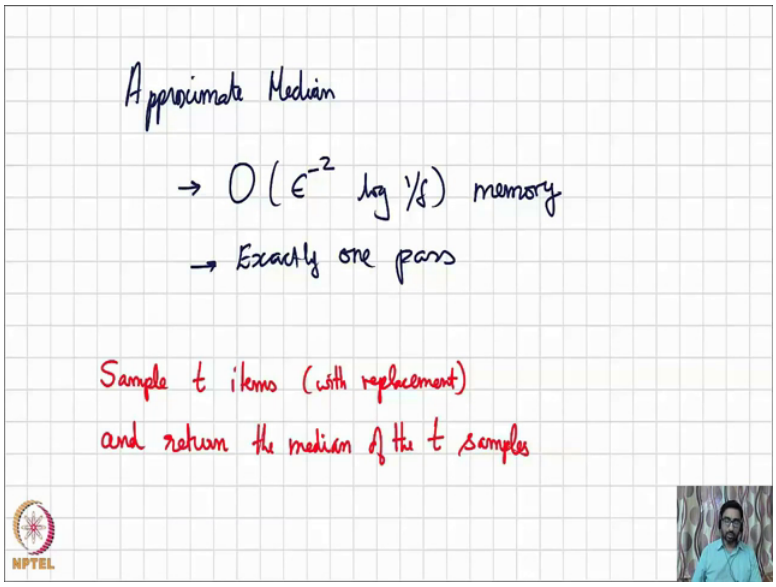
Just to remind ourselves we have introduced the notion of Streaming, we have also looked at Approximate Counting that requires just O of log log n bits, and we have looked at Sampling k items out of the n item stream when we do not know n; both with and without replacement. This is called Reservoir Sampling.

We are now ready to apply our ideas to finding an approximate median. Just recall, we already looked at the problem of finding the exact median, it requires multiple passes. We look direct from the classical RAM moral perspectives, and if you recall we had to sample some n to the three-fourths number of items, we have to find the median, and then we had to step back from median.

Basically, find two items within the samples, then go through the passes again find items that fell within the two items in the sample so and so forth. So, that requires a little bit more work requires multiple passes, little requires into the three-fourths memory and so

on so forth. It still works as a streaming algorithm, but you will need multiple passes and you will need into the three-fourths memory. So, what we are going to look at today is requires a lot lockless memory and just a single pass.

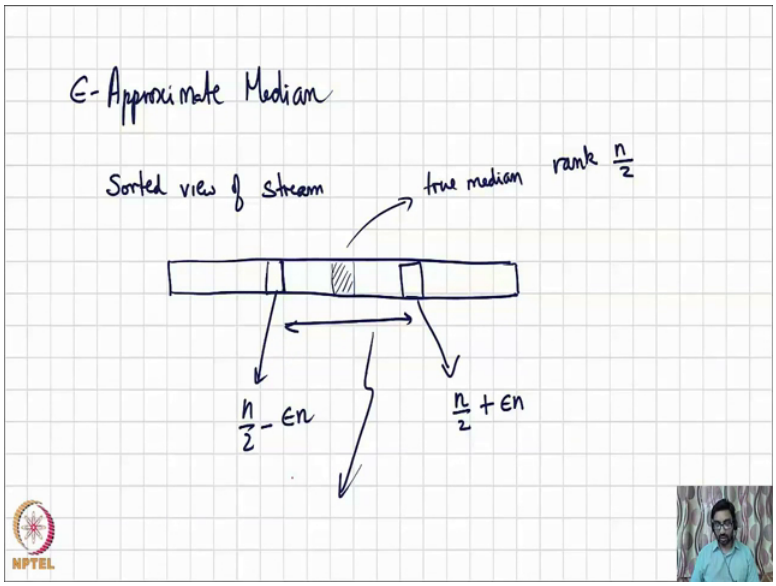
(Refer Slide Time: 02:07)



In particular, this approximate median algorithm will only requires O ($\varepsilon^{-2}\log\left(\frac{1}{\delta}\right)$) amount of memory and it will require exactly 1 pass. The algorithm itself is very simple. All we have to do is sample some t items with replacement from the stream and we know how to do that. And we need to simply the return the median of those t samples. So, it is probably the most natural algorithm that you can think of.

Of course, the only thing that remains is for us to know what the value of t is, and that is going to be our primary concern. If we plug in the appropriate value of t, we will get an approximate median, but for that we first need to be very clearly about what we mean by an approximate median.

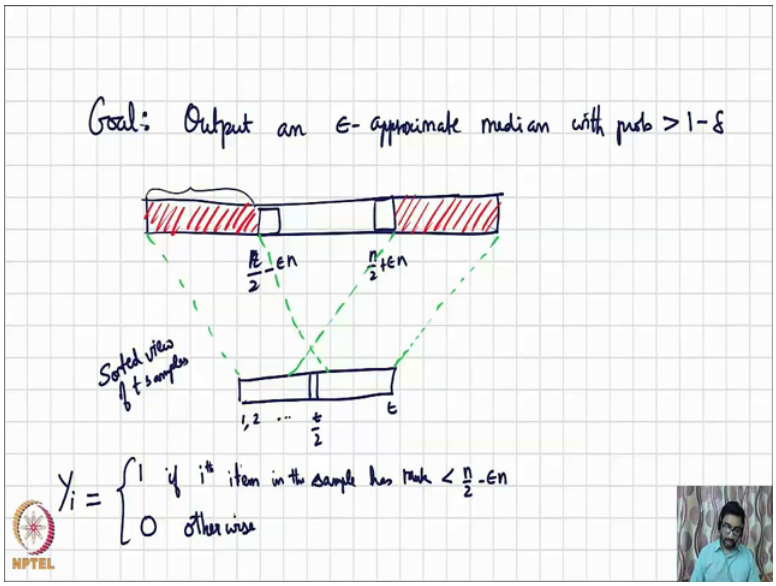
(Refer Slide Time: 03:11)



We all know what the true median of a set of an item is. You take the sorted view and the true median is right in the middle. It is simple the item with rank n of two and by rank here we mean the position in the sorted view. Now the question is what is an approximate median? And notice that we are requiring an epsilon approximate median so there is parameter.

So, what is an epsilon approximate median? Well, consider this range, this item is a rank $\frac{n}{2} - \varepsilon n$ and the item to the right is of rank $\frac{n}{2} + \varepsilon n$. Now any item within this range of rank $\frac{n}{2} - \varepsilon n$ up to rank $\frac{n}{2} + \varepsilon n$ is an epsilon approximate median.

(Refer Slide Time: 04:37)



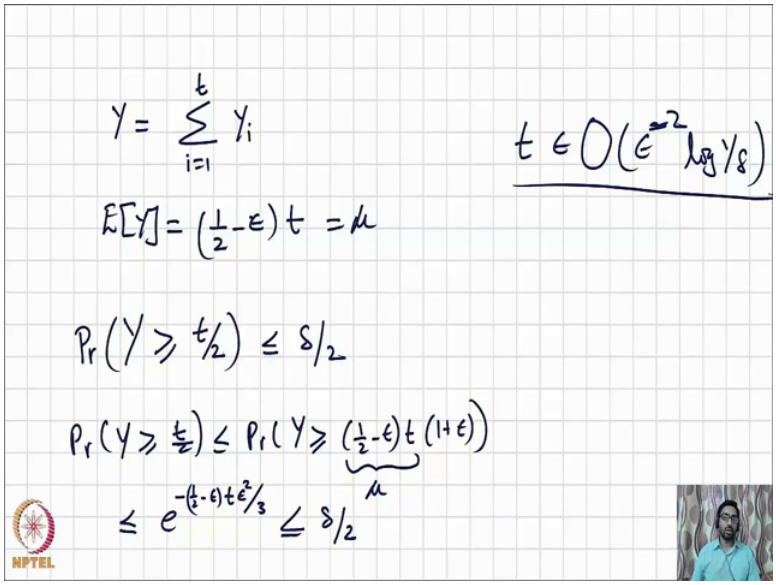
Now, that we know what an epsilon approximate median is, now let us said about finding a approximate t value for which our algorithm will output and epsilon approximate median with probability at least 1-δ. So, goal is to output an epsilon approximate median with probability at least 1-δ. So, how will this not happens that is the important question. Let us again look at the sorted view. In this sorted view, we certainly do not want to output an item that is shaded red, because that is falling outside of the acceptable range for an epsilon approximate median.

In particular, when we look at the sorted view of the t samples, there are two bear events that we want to avoid. We want to avoid getting more than t by 2 items or t by 2 or more items from the red range in the left. And similarly, we also want to avoid getting more than t by 2 items from the red range in the right. These are the two bad events and we want to avoid them. So, the question now is what is an appropriate value of t such that these two bad events can be avoided.

So for that, let us define random variable. Y i is our random variable and it is equal to 1 if the ith item in the sample now you just think of the sample has being ordered in some way and not necessarily sorted and the Y i equal to one if the ith item in the sample has rank less than $\frac{n}{2}-\varepsilon n$. Basically, this captures the event that the ith item is coming

from the red range on the left. So, such items, we do not want too many such items. So, with that being the case.

(Refer Slide Time: 06:53)



Let us define Y to be the summation over all t, over all i ranging from 1 to t Y i. And of course, we know that expectation of Y i equals to half minus epsilon t. Our bad event, the event that t by 2 or more items in the sample are coming from this red range in the left can be captured in the following way, that simply the event y is greater than or equal to t by 2. And of course, we want the probability of this event to be small. In particular we want this probability to be no more than delta by 2.

Why delta by 2? Because, there is a mirror image asymmetric other bad event that we also want to avoid the probability delta by 2, so the two bad events for together using the union bound, we can say that there the bad events do not occur with probability or it occur with probability at most delta and therefore the final outcome is correct with probability at least 1 minus delta. So, that is the goal. And now notice that these Y i's are independent. So, applying Chernoff bounds we look at the probability Y greater than or equal to t by 2 and we rewrite it as probability Y greater than or equal to half minus epsilon times t and that is mu times 1 plus epsilon.

This is smaller than $e^{-\left(\frac{1}{2}-\varepsilon\right)t\varepsilon^2/3}$, and we want this whole probability to be less than or equal to delta by 2. So, if you work out the mathematics we can include that this in equal

to we can be achieved with t belonging to O (ε - $\log\left(\frac{1}{\delta}\right)$¿ . In fact, I would encourage you to find out the exact t value for which this inequality holds.

So, with that we conclude our lecture on introducing streaming, looking at the problem of counting the number of items in a stream in a single pass using very small memory, the problem of sampling items, some k items from a stream both with or without replacement. Finally, now we have looked at the problem of finding the median and approximate median from this stream using very small amount of memory in particular O of epsilon to the minus 2 times log 1 over delta memory in a single pass to find an epsilon approximate median in our stream.